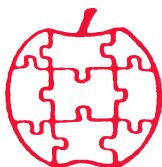


Apple

\$1.80



Assembly Line

Volume 7 -- Issue 9

June, 1987

Selective CLEAR for Applesoft Variables.	2
More Than Just Saving the Registers.	12
Correction to ProDOS Patcher	18
Using S-C Macro Assembler with Echo Speech Synthesizer . .	19
Assembly Listings on ProDOS Text Files	22
Printing a Tiny Address Booklet.	24

Return of Beneath Apple ProDOS

We've just received our copies from the newest printing of Beneath Apple ProDOS, so that excellent reference is available again. Our thanks to Quality Software for keeping such valuable information at our fingertips.

However, we do not expect that Quality Software will reprint either Understanding the Apple II or Understanding the Apple //e, so once the supplies run out that's the end of those books. If you are at all interested in the innermost details of how your Apple hardware works you need to get one of these outstanding books by Jim Sather. We've never seen anything to compare to these books for the wealth of information at the chip-by-chip level. See the S-C Software ad on page three for our price.

Latest from Lancaster

Don Lancaster just sent a copy of his latest book, Ask the Guru. Computer Shopper readers will recognize the title of his monthly column in that publication; this book is a collection of the columns from March, 1985 through March, 1987. Here are all of Don's writings on the inner workings of our favorite computer, especially how to get the most out of AppleWriter and/or the LaserWriter printer.

This is undoubtedly the best-printed self-published book I've ever seen: each copy is directly printed out on Don's LaserWriter! Even subscribers to Computer Shopper will want this book, just to have all the information in one place rather than strung out through many pages of oversized newsprint. \$24.50 from Synergetics, Box 809, Thatcher, AZ 85552. (602) 428-4073.

Selective CLEAR for Applesoft Variables.....Bruce E. Howell

A friend of mine wrote an Applesoft program that processes classroom grades. Certain information, such as the teacher's name, would be loaded and used throughout the program, while other information, such as student grades, had to be cleared often. One solution would be having a loop to clear certain variables. This would be quick and easy with simple variables, but with multi-dimensioned arrays this can be time consuming. Another solution would be to open a temporary file and store the static information there or to POKE it in a safe place in memory, such as at \$300, issue the CLEAR command to clear all Applesoft variables, then re-load the variables. My solution to the problem is Selective Clear, which allows you to select a range of real, integer or string variables to clear, or to clear an entire, specific array. Note that Selective Clear sets real and integer variables to zero, and string variables to a null string. Applesoft's CLEAR command deletes the variables, requiring arrays to be re-defined after the CLEAR.

USING SELECTIVE CLEAR

Selective Clear works with either ProDOS or DOS. To follow ProDOS convention, it is loaded at \$2000, then moved up in memory to where it is used. If your Applesoft program is short, you can load Selective Clear from within your program. A longer Applesoft program would be overwritten by the initial loading of Selective Clear. To avoid this, use a short Applesoft program to load Selective Clear, then run the main program as follows:

```
10 PRINT CHR$(4)"BRUN B.SEL.CLEAR"  
20 PRINT CHR$(4)"RUN MAIN.PROGRAM"
```

To use Selective Clear in your application, it is best to define your simple variables and group them alphabetically so that those that you want to clear often are all at one end or the other of the alphabet. The syntax for simple variables is:

```
& $ > G : REM clear all string variable greater than G  
& % < M : REM clear all integer variables less than M  
& R > T : REM clear all real variables greater than T
```

Following the ampersand, use \$, % or R to indicate string, integer or real variables to clear. Following the variable type use the greater than (>) or less than (<) sign, then a single alpha character to show the range.

Array variables are cleared individually by placing the actual variable to clear in parenthesis following the ampersand as follows:

```
& (AB$) : REM To clear the string array AB$  
& (M) : REM To clear the real array M  
& (TT%) : REM To clear the integer array TT%
```

Note that if you try to clear an array that has not been defined, you will get an error message of ARRAY ERROR.

S-C Macro Assembler Version 2.0.....DOS \$100, ProDOS \$100, both for \$120
Version 2.0 DOS Upgrade Kit for 1.0/1.1/1.2 owners.....\$20
ProDOS Upgrade Kit for Version 2.0 DOS owners.....\$30
Source Code of S-C Macro 2.0 (DOS or ProDOS).....each, additional \$100
S-C DisAssembler (ProDOS only).....without source code \$30, with source \$50
RAK-Ware DISASM (DOS only).....without source code \$30, with source \$50
ProVIEW (ProDOS only).....\$20
Full Screen Editor for S-C Macro (with complete source code).....\$49
S-C Cross Reference Utility.....without source code \$20, with source \$50
S-C Word Processor (with complete source code).....\$50
DPL8 and DPPP, including complete source and object code.....\$50
ES-CAPE (Extended S-C Applesoft Program Editor).....
Including Version 2.0 With Source Code.....\$50
ES-CAPE Version 2.0 and Source Code Update (for Registered Owners)....\$30
Bag of Tricks 2 (Quality Software).....(\$49.95) \$45 *
MacASM -- Macro Assembler for Macintosh (Mainstay).....(\$150.00) \$50 *
S-C Documentor (complete commented source code of Applesoft ROMs).....\$50
Cross Assemblers for owners of S-C Macro Assembler.....\$32.50 to \$50 each
(Available: 6800/1/2, 6301, 6805, 6809, 68000, Z-80, Z-8, 8048,
8051, 8085, 1802/4/5, PDP-11, G1650/70, others)
Beagle Bros. Applesoft Compiler (ProDOS only).....(\$74.95) \$65 *

AAL Quarterly Disks.....each \$15, or any four for \$45
Each disk contains the source code from three issues of AAL:
Jan-Mar, Apr-Jun, Jul-Sep, and Oct-Dec.
(All source code is formatted for S-C Macro Assembler. Other assemblers
require some effort to convert file type and edit directives.)

Diskettes (with hub rings)..... package of 25 for \$15 *
Vinyl disk pages, 6"x8.5", hold two disks each.....10 for \$6 *
Diskette Mailing Protectors (hold 1 or 2 disks).....40 cents each
(Cardboard folders designed to fit 6"x9" Envelopes.) or \$25 per 100 *
Envelopes for Diskette Mailers..... 6 cents each

65802 Microprocessor (Western Design Center)..... \$25 *
quikLoader EPROM System (SCRG).....(\$179) \$170 *
PROMGRAMMER (SCRG).....(\$149.50) \$140 *
Minuteman 250 Uninterruptible Power Supply.....(\$359) \$320 +

"Apple //gs Technical Reference", Fischer.....(\$19.95) \$19 *
"65816/65802 Assembly Language Programming", Fischer.....(\$21.95) \$20 *
"Programming the 65816", Eyes.....(\$22.95) \$21 *
"Apple //e Reference Manual", Apple Computer.....(\$24.95) \$23 *
"Apple //c Reference Manual", Apple Computer.....(\$24.95) \$23 *
"ProDOS Technical Reference Manual", Apple Computer.....(\$29.95) \$27 *
"ProDOS Inside and Out", Doms & Weishaar.....(\$16.95) \$16 *
"DOSTalk Scrapbook", Weishaar & Kersey.....(\$14.95) \$14 *
"Beneath Apple ProDOS", Worth & Lechner.....(\$19.95) \$18 *
"Beneath Apple DOS", Worth & Lechner.....(\$19.95) \$18 *
"Inside the Apple //c", Little.....(\$19.95) \$18 *
"Inside the Apple //e", Little.....(\$19.95) \$18 *
"Understanding the Apple //e", Sather.....(\$24.95) \$23 *
"Understanding the Apple II", Sather.....(\$22.95) \$21 *
"Apple II+/Ile Troubleshooting & Repair Guide", Brenner.....(\$19.95) \$18 *
"Apple II Circuit Description", Gayler.....(\$22.95) \$21 *
"Assem. Language for Applesoft Programmers", Finley & Myers.....(\$18.95) \$18 *
"Now That You Know Apple Assembly Language...", Gilder.....(\$19.95) \$18 *
"Enhancing Your Apple II, vol. 1", Lancaster.....(\$15.95) \$15 *
"Enhancing Your Apple II, vol. 2", Lancaster.....(\$17.95) \$17 *
"Assembly Cookbook for the Apple II/Ile", Lancaster.....(\$21.95) \$20 *
"Microcomputer Graphics", Myers.....(\$14.95) \$14 *
"Assembly Lines -- the Book", Wagner.....(\$19.95) \$18 *
"AppleVisions", Bishop & Grossberger.....(\$39.95) \$36 *

* On these items add \$2.00 for the first item and
\$.75 for each additional item for US shipping.

+ Inquire for shipping cost.

Foreign customers inquire for postage needed.

Texas residents please add 7.25 % sales tax to all orders.

*** S-C SOFTWARE, P. O. BOX 280300, Dallas, TX 75228 ***

*** (214) 324-2050 ***

*** Master Card, VISA, Discover and American Express ***

ENTERING THE PROGRAM

To enter Selective Clear, you can use an assembler to type in the source code as listed. I use the ProDOS version of S-C Macro Assembler. You can also enter it directly into the monitor. If you do the latter, save it with the command:

```
BSAVE B.SEL.CLEAR,A$2000,L$166
```

HOW SELECTIVE CLEAR WORKS

The first thing done is to identify which operating system is in effect, to properly reserve memory for Selective Clear. Then the routine is moved to the proper location in memory, the ampersand vector is set and control returns back to Applesoft.

To understand how simple variables can be selected and cleared, you must understand how variables are stored in memory. Locations \$69,6A point to the start of a table of simple variables, normally set to LOMEM. For real and integer variables, the actual values are stored in the table, while for string variables the table contains pointers to where the strings are stored in memory (from HIMEM down). Each variable occupies seven locations or bytes in the table, the first two locations being the variable name.

With real variables, the first byte is the positive value of the first character of the variable name. The second byte is the positive value of the second character of the variable name. For integer variables, both bytes are the negative values for the variable name, while with string variables, the first byte is positive and the second is negative. Depending on which variable type is selected, two different masks (MASK1 and MASK2) are used to determine if the current entry in the table has a variable name with the proper signs for the type you want to select.

The other choice that Selective Clear needs to make is whether the variable in the table is less than or greater than the range that you have indicated. This is done by changing the program slightly each time it is called. If greater than is selected, then a Branch on Carry Clear (BCC) is placed where the variable name is compared to the selected range. Greater than stores a Branch on Carry Set (BCS). A no match of either the variable type or range skips to the next entry in the table until the end of the table is reached.

If the variable name is within the selected range and the signs match the variable type selected, then Selective Clear clears the variable by storing zeros in the next five locations in the table. For real numbers, the first location contains a one byte exponent followed by four bytes of mantissa. Integer numbers have the high byte first followed by the low byte of the number stored in the variable. The last three bytes of the entry are not used. String variables have a one byte string length followed by the low and high bytes of the address where the string is stored. With strings, the last two bytes of the table entry are not used.

Memory locations \$6B,6C point to the start of the array table. Variable names are stored in the same way as with simple variables. Since you will manually select which array to clear, the routine finds the variable's location in the table by evaluating the expression within the parenthesis using an Applesoft routine at \$DEB2, then searching the table for the variable name. The first two bytes following the name are an offset pointer to the next variable in the table. Selective Clear uses their value to determine when it is done clearing. The next byte is the number of dimensions present. It uses that value to bypass that number of sets of high-low bytes which follow, each set indicating the size of each dimension. After these sets of bytes is where the actual data is stored (or pointers to data if string arrays) in the same way that simple variables are stored. To clear, the routine zeros all locations till it reaches the next array variable name.

CUSTOMIZING THE ROUTINE

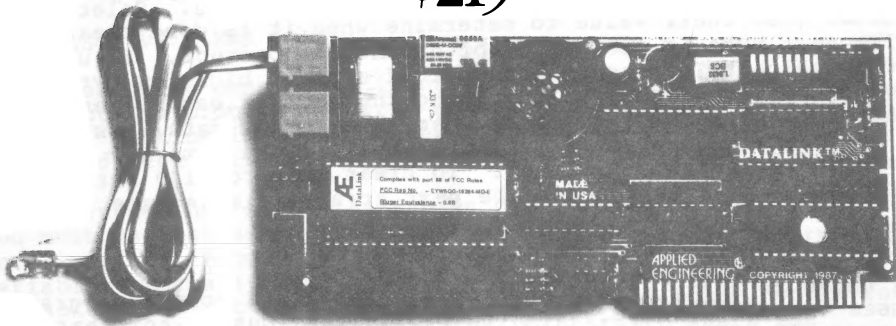
I chose to clear arrays individually, since a normal program uses very few arrays. You can modify the routine to clear a range of arrays using the technique outlined for simple variables. If you need to clear either simple variables only or arrays only, you can shorten the routine enough to fit at \$300. You could also modify the routine to set the variable to some required value, passed to the routine, instead of clearing it.

```

1000 *SAVE S.SEL.CLEAR
1010 *-----
1020 *
1030 * Selective CLEAR
1040 *
1050 * Copyright 1987 by Bruce E. Howell, D.D.S.
1060 * 2531 Cartwright Rd.
1070 * Missouri City, Texas 77459
1080 *
1090 * Format for simple variables is: &type>range
1100 * &type<range
1110 *
1120 * Where 'type' is: $ to clear string variables
1130 * % to clear integer variables
1140 * R to clear real variables
1150 *
1160 * 'range' is a single alpha character.
1170 * Variables greater (>) or less (<) than
1180 * this character are cleared.
1190 *
1200 *
1210 * Format for array variables is:
1220 *
1230 * & (C$) to clear the string array C$
1240 * & (C) to clear the real array C
1250 * & (C%) to clear the integer array C%
1260 *
1270 *-----
B1- 1280 CHRGET .EQ $00B1
B7- 1290 CHRGOT .EQ $00B7
03F5- 1300 AMPER .EQ $03F5 Ampersand Vector
1310 *-----
BEF8- 1320 FREBUF .EQ $BEF8 PRODOS FREE BUFFERS
BEF5- 1330 GETBUF .EQ $BEF5 PRODOS Obtain Buffer
1340 *-----
D412- 1350 ERROR .EQ $D412 PRINT ERROR
D998- 1360 ADDON .EQ $D998 Add (Y) to ChrPtr
D9A3- 1370 SCAN .EQ $D9A3 SCAN TO :/EOL
DEB2- 1380 PARCHK .EQ $DEB2 EVAL (EXPRES)
1390 *-----
FE2C- 1400 MOVMEM .EQ $FE2C MOVE MEMORY
1410 *-----

```

Global communications. Down-to-earth price. \$219



Introducing the Datalink™ modem.
Get instant access to networks and databases
— plus network memberships and discounts worth \$177.95!

Now you can tap into a world of information — without draining your resources. Our Datalink™ modem lets your Apple IIGS, IIe or II+ communicate with other computers, download free software from networks and bulletin boards, access database services and more. For a remarkably low price.

The world at your fingertips.

With the Datalink modem, you'll be able to draw information from thousands of databases. Send and receive electronic mail — even overseas. Join clubs and bulletin boards to exchange software or solve computing problems with other Apple users. Download free public domain software. And share files with personal computers and mainframes.

There's nothing missing on Datalink.

Unlike the Smartmodem®, Datalink comes with its own easy-to-use communications software in ROM and on disk that supports macros, file transfers, on-line time display, data capture and datascopes mode. The software also allows you to store hundreds of phone numbers for auto dialing and log on.

Because Datalink is 100% Super Serial Card compatible, you can also run virtually all other communication software, including Point-to-Point, Apple Access and ASCII Express, to name a few.

The compact design allows Datalink to fit in any slot — including slots 1 and 2 of the IIGS with a cooling fan installed. Datalink operates at 1200 or 300 baud. Built-in diagnostics check for accuracy of data transmission, and Datalink lets you track the progress of calls either electronically or via an on-board speaker.

Introductory offers from popular networks included.

When you purchase Datalink, you'll get a fee-waived membership to The Source worth \$49.95, \$60.00 worth of free on-line time from News Net. A free \$50.00 subscription to the Official Airline Guide. A free subscription to the GENIE network worth \$18.00! And information on MCI Mail.

Order today!

To order or for more information, see your dealer or call Applied Engineering today, 9 a.m. to 11 p.m. 7 days. Or send check or money order to Applied Engineering. MasterCard, VISA and C.O.D. welcome. Texas residents add 6 1/4% sales tax. Add \$10.00 outside U.S.A.

	Applied Engineering Datalink	Hayes® Smartmodem 1200A
Price	\$219	\$439
Max. transmission rate	1200 baud	1200 baud
Warranty period	5 years at no charge	2 years or 4 years for \$75
Software included	YES	NO
Hayes AT command set	YES	YES
Help screens	YES	NO
On-board telephone jacks	YES	NO
Fits any slot (even with fan)	YES	NO

Smartmodem is a registered trademark of D.C. Hayes Corporation

AE Applied Engineering
The Apple enhancement experts.

Box 798, Carrollton, TX 75006
(214) 241-6060

```

3C-      1420 SSA      .EQ $3C,3D  SOURCE START ADDRESS
3E-      1430 SEA      .EQ $3E,3F  SOURCE END ADDRESS
42-      1440 DSA      .EQ $42,43  DESTINATION START ADDRESS
      1450 *-----
69-      1460 VARTAB   .EQ $69,6A  START OF VARS
6B-      1470 ARYTAB   .EQ $6B,6C  START ARRAYS
6D-      1480 STREND   .EQ $6D,6E  END OF ARRAY
6F-      1490 FRETOP   .EQ $6F,70
73-      1500 HIMEM    .EQ $73,74
81-      1510 VARNAM   .EQ $81,82
E0-      1520 LOC      .EQ $E0,E1
E8-      1530 HOLD     .EQ $E8,E9
E4-      1540 RANGE    .EQ $E4
E5-      1550 MASK1    .EQ $E5
E6-      1560 MASK2    .EQ $E6
      1570 *-----
4C-      1580 JMP.OP   .EQ $4C      JMP OP CODE
80-      1590 MASK     .EQ $80
90-      1600 BCC.OP   .EQ $90      BCC OP CODE
B0-      1610 BCS.OP   .EQ $B0      BCS OP CODE
      1620 *-----
      1630          .MA SET
      1640          LDA #J2
      1650          STA J1
      1660          LDA J12
      1670          STA J1+1
      1680          .EM
      1690 *-----
      1700          .MA MOV
      1710          LDA J2
      1720          STA J1
      1730          LDA J2+1
      1740          STA J1+1
      1750          .EM
      1760 *-----
      1770          .OR $2000
      1780          .TF B.SEL.CLEAR
      1790 *-----
2000-    1800 INIT     LDA $BFOO    MLI LOCATION
2003- C9 4C      1810 CMP #JMP.OP  SEE IF ACTIVE
2005- F0 0E      1820 BEQ PRODOS   YES DO PRODOS
      1830 *---Assume in DOS 3.3-----
      1840 DOS         LDA #SELCLR
      1850             STA HIMEM
      1860             STA FRETOP
      1870             LDA /SELCLR
      1880             STA HIMEM+1
      1890             STA FRETOP+1
      1900             BNE DOMOVE    ALWAYS BRANCH
      1910 *---Assume in ProDOS-----
2015- 20 F8 BE   1920 PRODOS JSR FREBUF  FREE ALL BUFFERS
2018- A9 02      1930         LDA #2    GET TWO PAGES
201A- 20 F5 BE   1940         JSR GETBUF
      1950 *---Move routine into hole-----
201D-    1960 DOMOVE >SET SSA,FROM  Starting address of source
201D- A9 48      0000>         LDA #FROM
201F- 85 3C      0000>         STA SSA
2021- A9 20      0000>         LDA /FROM
2023- 85 3D      0000>         STA SSA+1
2025-    1970      0000>         >SET SEA,THRU  Ending address of source
2025- A9 66      0000>         LDA #THRU
2027- 85 3E      0000>         STA SEA
2029- A9 21      0000>         LDA /THRU
202B- 85 3F      0000>         STA SEA+1
202D-    1980      0000>         >SET DSA,SELCLR Starting address of destination
202D- A9 00      0000>         LDA #SELCLR
202F- 85 42      0000>         STA DSA
2031- A9 98      0000>         LDA /SELCLR
2033- 85 43      0000>         STA DSA+1
2035- 20 2C FE   1990      0000>         JSR MOVMEM
2038- A9 4C      2000      0000>         LDA #JMP.OP  SET UP AMPERSAND
203A- 8D F5 03   2010      0000>         STA AMPER
203D-    2020      0000>         >SET AMPER+1,SELCLR
203D- A9 00      0000>         LDA #SELCLR
203F- 8D F6 03   2030      0000>         STA AMPER+1
2042- A9 98      0000>         LDA /SELCLR
2044- 8D F7 03   2040      0000>         STA AMPER+1+1
2047- 60      2050      0000>         RTS          RETURN
      2040 *-----
      2050 FROM

```

```

9800- 20 B7 00 2060 .PH $9800 WHERE WE END UP
9803- A0 80 2070 SELCLR JSR CHRGOT Get char after "&"
9805- A2 80 2080 LDY #MASK If "R": X=$80, Y=$80
9807- C9 52 2090 LDX #MASK If "$": X=$80, Y=$00
9809- F0 0F 2100 CMP #'R' If "%": X=$00, Y=$00
980B- C9 24 2110 BEQ .3 ...REAL VARIABLE
980D- F0 09 2120 CMP #'$' SEE IF STRING
980F- C9 25 2130 BEQ .2 ...STRING VARIABLE
9811- F0 03 2140 CMP #'%' SEE IF INTEGER
9813- 4C A6 98 2150 BEQ .1 ...INTEGER VARIABLE
9816- A2 00 2160 JMP SEARCH.ARRAY.VARIABLES TRY ARRAY
9818- A0 00 2170 .1 LDX #000 CHECK FOR NEG
981A- 86 E5 2180 .2 LDY #000
981C- 84 E6 2190 .3 STX MASK1 SET MASKS FOR
          2200 STY MASK2 TWO CHAR. NAME
          2210 *---Get and save relop-----
981E- 20 B1 00 2220 JSR CHRGET GET FUNCTION <>
9821- 8D 66 98 2230 STA FUNC
          2240 *---Get and save letter-----
9824- 20 B1 00 2250 JSR CHRGET GET RANGE OF Variable Name
9827- 85 E4 2260 STA RANGE
          2270 *---Modify code for relop-----
9829- AD 66 98 2280 LDA FUNC
982C- C9 CF 2290 CMP #$CF SEE IF GT >
982E- D0 06 2300 BNE .4 ...not >
9830- E6 E4 2310 INC RANGE SO > RANGE NOT =
9832- A9 90 2320 LDA #BCC.OP SET UP GT >
9834- D0 0B 2330 BNE .6 ...always
9836- C9 D1 2340 .4 CMP #$D1 SEE IF LT <
9838- F0 05 2350 BEQ .5 ...it is <
983A- A2 10 2360 LDX #$10 REPORT SYNTAX ERROR
983C- 4C 12 D4 2370 JMP ERROR
983F- A9 B0 2380 .5 LDA #BCS.OP SET UP LT <
9841- 8D 66 98 2390 .6 STA FUNC FUNNY TRICK
          2400 *-----
          2410 SEARCH.SIMPLE.VARIABLES
9844- A0 00 2420 LDY #0 FOR INDEX
9846- 2430 >MOV LOC,VARTAB START OF VARIABLE TABLE
9848- A5 69 0000> LDA VARTAB
984A- 85 E0 0000> STA LOC
984C- A5 6A 0000> LDA VARTAB+1
984E- 85 E1 0000> STA LOC+1
9850- C5 E1 2440 SIMPLE LDA LOC+1 ARE WE DONE?
9852- D0 0C 2450 CMP ARYTAB+1
9854- A5 E0 2460 BNE .1
9856- C5 06 2470 LDA LOC
9858- 90 0B 2480 CMP ARYTAB
985A- 20 A3 D9 2490 BCC .1
985D- 4C 98 D9 2500 JSR SCAN YES SCAN EOL
          2510 JMP ADDON GO APPLIESOFT
          2520 *-----
9860- B1 E0 2530 .1 LDA (LOC),Y GET VARIABLE NAME
9862- 29 7F 2540 AND #$7F MASK HIGH BIT
9864- C5 E4 2550 CMP RANGE COMPARE TO RANGE
          2560 *
9866- 90 08 2570 FUNC BCC .1 ***Opcode modified to BCC or BCS***
          2580 *
9868- B1 E0 2590 LDA (LOC),Y GET VARIABLE TYPE
986A- 29 80 2600 AND #MASK
986C- 45 E5 2610 EOR MASK1
986E- D0 13 2620 BNE .3
9870- 20 9F 98 2630 .1 JSR INCLOC
9873- 20 9F 98 2640 JSR INCLOC
9876- A5 E0 2650 .2 LDA LOC SKIP FIVE
9878- 18 2660 CLC
9879- 69 05 2670 ADC #$05
987B- 85 E0 2680 STA LOC
987D- 90 CF 2690 BCC SIMPLE
987F- E6 E1 2700 INC LOC+1
9881- B0 CB 2710 BCS SIMPLE ALWAYS
          2720 *-----
9883- 20 9F 98 2730 .3 JSR INCLOC
9886- B1 E0 2740 LDA (LOC),Y
9888- 29 80 2750 AND #MASK
988A- 45 E6 2760 EOR MASK2
988C- F0 E8 2770 BEQ .2 ...not correct type
          2780 *---Clear the value-----
988E- 20 9F 98 2790 JSR INCLOC PASS 2ND
9891- A2 05 2800 LDX #5
9893- A9 00 2810 LDA #0 Store zero in 5 bytes

```


17 Meg IIfx/1 Meg IIfx

Want the best & easiest-to-use (most support) RAM card on the market — at the best price? Buy Checkmate Technology's **State-Of-The-Art MULTIRAM RGB RAM CARD™** from us w/**BATTERY BACKED-UP RAM** options that save programs like **AppleWorks** for years! **FAST CHEAP REPLACEMENT FOR HARD DISKS, USER EXPANDABLE TO 17 MEGS**, (100%) compatible w/all 3rd party software/hardware, 65C816 option (\$147), sharp 80 columns/Double Hi-Res, & **FREE RGB!** Direct substitute for Ramworks I-III™ (we think it's better) by Applied Engineering (also known as Preferred Computing) or Apple 80 col cards, has a **5 YR WARRANTY, 15 day money back guarantee, & FREE AUTOMATIC SOFTWARE UPDATES ONLY FROM US!**

MultiRam RGB expands to 1 Meg + optional **BATTERY PROTECTABLE MultiRam Plus** piggyback expandable to 16 Meg's. **MultiRam IIfx** expands to 768k. The software & manual is the best & easiest-to-use on the market — without exception!

FREE APPLEWORKS EXPANDER SOFTWARE W/EVERY CARD — loads ALL versions of Appleworks (in 30 seconds w/ AW 2.0), even printer routines, runs faster, Desktop 2048k, 23,000+ records & 4,800+ word proc lines, segments lg files onto multiple disks! **FREE AW TIME/DAY/DATE ON-SCREEN** w/ clock, **AUTO-COPY TO RAM, PRODOS/DOS 3.3. RAM DISK & TEST.** Optional CP/M & Pascal Ram disks (\$20/ea)!

	MultiRam RGB Card	MultiRam IIfx Card
64k MultiRam	159. . . .	129. . . .
320k MultiRam	189. . . .	157. . . .
576k MultiRam	219. . . .	187. . . .
832k MultiRam	244. . . .	212. . . .
1024k MultiRam	269. . . .	← ← ← ←
	MultiRam Plus Piggyback	
256k MultiRam Plus	219. . . .	NEW
512k MultiRam Plus	269. . . .	NEW
768k MultiRam Plus	316. . . .	NEW
1024k MultiRam Plus	359. . . .	NEW
2-17 meg MultiRam Plus	Call	NEW
Battery/AC Kit (for Plus)	36. . . .	NEW

WHY BUY FROM COIT VALLEY COMPUTERS RATHER THAN SOME MAIL ORDER HOUSES? Only we offer a **15 day MultiRam complete money back satisfaction guarantee**, extra software, more support, **free automatic software updates**, free 64k w/ each 256k/512k/768k IIfx/RGB card. We know the products, & stock them for fast shipping! **CALL FOR SUPPORT, QUESTIONS, PRICES, DISCOUNTS. SCHOOLS & GROUPS WELCOME.**

Checkmate Technology's MultiRam CX card easily expands your IIfx to 640k and has a CX + Piggyback to add another 512k (1152k total)! It's 100% compatible with all IIfx software/hardware & upgradable with a 65C816 kit (\$95) — all at about 50% less power than Z-RAM™! Has the SAME **FREE SOFTWARE/UPDATES & 5 YR WARRANTY AS IIfx CARDS. Buy a C-VUE flat panel display for \$287 w/ 512k or greater card.**

256k MULTIRAM CX	155.
512k MULTIRAM CX	184.
768k MULTIRAM CX Combo	328.
1024k MULTIRAM CX Combo	354.
256k MULTIRAM CX+	199.
512k MULTIRAM CX+	209.
1200 Baud IIfx Modem (add \$20 for cable)	147.
Clip-on IIfx Rechargeable Battery	115.
IIfx Flat Panel Display (w/ 512k > Mram)	287.
IIfx System Clock	69.

Cermetek 1200 internal modem IIfx/II+/IIfx 148.
(+ **FREE** Source subscription/ VT 100 Shareware/ ASCII Express & ProTERM™ Setup Instructions)

SLOT 1-7 Ram Card for II+, IIfx, Laser	Call
ProAPP 20 Meg Hard Disks (+ FREE software)	Call
Apple IIfx Enhancement Kit	62.
Apple IIfx	Call
Apple 3.5 800k Unidrive (IIfx/IIfx+/IIfx)	319.
Super Serial Compatible Card	87.
RamPak 512k qs RAM card	196.
Apple Compatible Drive IIfx/II+ (IIfx/IIfx+ \$19)	132.
Clockworks Clock (Thunder/TimeHO™ compat)	84.
13" RGB Monitor w/ cables (req. RGB card)	327.
Heavy Duty Power Supply II/II+/IIfx	69.
IIfx Programmable Numeric Keypad	85.
System Saver Fan (II/II+/IIfx)	66.
Uninterruptable Power Supply 250 watt	299.
AutoWorks (Best AW Macro + more)	35.
FontWorks (Best AW sideways/Fonts & more)	35.
Pinpoint Program	55.
Pinpoint Document Checker (Best Speller)	45.
Copy II Plus 7.3 (Best ProDOS Utility)	29.
ProTERM Modem Software	84.
ASCII Express Modem Software	84.
VIP Professional (Lotus Clone. Specify Comp)	169.

Terms: Add \$4-Ground or \$6-Air shipping & phone # to each U.S. MultiRam card order (foreign orders/FPO/APO extra). Add 3% for MasterCard/Visa/Am Exp (include expir date) & P.O.'s. For fast delivery send Cashier's/Certified check/Money Order. C.O.D. (add \$5) & personal checks accepted (allow 16 days). Tex res add 7 1/2% tax. Software non-returnable.

Ordering: **PRINT** Name & address, Res & Bus phone, credit card name/number/expir date where appropriate, qty/description/price/shipping charges if known. Call for questions.

AutoWorks, Ramworks, II TimeMaster, II HO, Z-Ram, respective trademarks of Software Touch, Applied Engineering. Prices subject to change.

COIT VALLEY COMPUTERS
(214) 234-5047

MCI 2969684
Telex 6502969684 MCI UW

14055 Waterfall Way
Dallas, Texas 75240

```

9895- 91 E0 2820 .4 STA (LOC),Y
9897- 20 9F 98 2830 JSR INCLOC
989A- CA 2840 DEX
989B- D0 F8 2850 BNE .4
989D- F0 AF 2860 BEQ SIMPLE ...always
2870 *-----
989F- E6 E0 2880 INCLOC INC LOC
98A1- D0 02 2890 BNE .1
98A3- E6 E1 2900 INC LOC+1 ADVANCE HIGH
98A5- 60 2910 RTS
2920 *-----
2930 SEARCH.ARRAY.VARIABLES
98A6- 20 B2 DE 2940 JSR PARCHK ERROR IF NOT
98A9- 2950 >MOV LOC,ARYTAB START OF ARRAY TABLE
98A9- A5 6B 0000> LDA ARYTAB
98AB- 85 E0 0000> STA LOC
98AD- A5 6C 0000> LDA ARYTAB+1
98AF- 85 E1 0000> STA LOC+1
98B1- A0 00 2960 .1 LDY #0 FOR INDEX
98B3- B1 E0 2970 LDA (LOC),Y GET 1ST. CHAR.
98B5- F0 62 2980 BEQ .99 NONE PRESENT
98B7- C5 81 2990 CMP VARNAM SEE IF SAME NAME
98B9- D0 40 3000 BNE .5 FIND NEXT ARRAY
98BB- C8 3010 INY
98BC- B1 E0 3020 LDA (LOC),Y GET 2ND CHAR.
98BE- C5 82 3030 CMP VARNAM+1
98C0- D0 39 3040 BNE .5 FIND NEXT ARRAY
98C2- C8 3050 INY
98C3- B1 E0 3060 LDA (LOC),Y PASS 1ST NAME
98C5- 18 3070 CLC GET LOW OFFSET
98C6- 65 E0 3080 ADC LOC CLEAR CARRY
98C8- 85 E8 3090 STA HOLD ADD TO CURRENT
98CA- C8 3100 INY LOW END OF ARRAY
98CB- B1 E0 3110 LDA (LOC),Y GET HIGH OFFSET
98CD- 65 E1 3120 ADC LOC+1 ADD TO CURRENT
98CF- 85 E9 3130 STA HOLD+1 HIGH END OF ARRAY
98D1- C8 3140 INY
98D2- 20 9F 98 3150 .2 JSR INCLOC
98D5- 88 3160 DEY
98D6- D0 FA 3170 BNE .2
98D8- B1 E0 3180 LDA (LOC),Y # OF DIMS
98DA- AA 3190 TAX COUNTER
98DB- 20 9F 98 3200 JSR INCLOC PASS # DIMS
98DE- 20 9F 98 3210 .3 JSR INCLOC PASS SIZE H
98E1- 20 9F 98 3220 JSR INCLOC PASS SIZE L
98E4- CA 3230 DEX
98E5- D0 F7 3240 BNE .3
98E7- A9 00 3250 LDA #$00
98E9- 91 E0 3260 .4 STA (LOC),Y
98EB- 20 9F 98 3270 JSR INCLOC
98EE- A6 E0 3280 LDX LOC SEE IF END OF
98F0- E4 E8 3290 CPX HOLD THIS ARRAY YET
98F2- D0 F5 3300 BNE .4 NOT END OF ARRAY
98F4- A6 E1 3310 LDX LOC+1 CHECK AGAIN
98F6- E4 E9 3320 CPX HOLD+1
98F8- D0 EF 3330 BNE .4
98FA- 60 3340 RTS RETURN BASIC
3350 *---Find Next Array-----
98FB- A0 02 3360 .5 LDY #$02
98FD- B1 E0 3370 LDA (LOC),Y
98FF- 18 3380 CLC
9900- 65 E0 3390 ADC LOC
9902- 85 E8 3400 STA HOLD NEXT ADDRESS LOW
9904- C8 3410 INY
9905- B1 E0 3420 LDA (LOC),Y
9907- 65 E1 3430 ADC LOC+1
9909- 85 E1 3440 STA LOC+1 NEXT ADDRESS HIGH
990B- A5 E8 3450 LDA HOLD
990D- 85 E0 3460 STA LOC
990F- C5 6D 3470 CMP STREND
9911- D0 9E 3480 BNE .1
9913- A5 E1 3490 LDA LOC+1
9915- C5 6E 3500 CMP STREND+1
9917- D0 98 3510 BNE .1
3520 *
9919- A2 80 3530 .99 LDX #$80 DIDN'T FIND
991B- 4C 12 D4 3540 JMP ERROR say ARRAY ERROR
3550 .EP
3560 THRU

```

Æ Update . . .

AE's IIC System Clock now includes AppleWorks 2.0 support for time and date screen display and file stamping. This latest update now supports all versions of AppleWorks from 1.0 to 2.0. Using the IIC System Clock's pass-through serial port, the IIC clock can plug into either of the Apple IIC's built-in modem or printer ports. Then, the modem or printer can plug directly into the IIC System Clock. Three "AA" batteries keep the correct time for up to 2 years before simple replacement is necessary. Current IIC System Clock owners can upgrade to the new IIC System Clock Utilities, version 3.0.2 for \$10.00.

ColorLink, a new version of the RamWorks RGB option, is now shipping. ColorLink allows connection of an IBM 9-pin RGB monitor or the new Apple IIGs analog color RGB monitor to the Apple IIC. AE's previous version, called Digital Prism, supported the discontinued Apple Color 100 RGB monitor. ColorLink has the same \$129.00 retail price and allows the user a choice of viewing either white, blue, green or amber text. Digital Prism will still be available for Apple Color 100 monitor owners. Feel free to call Applied Engineering for advice on RGB monitors you can use with your system.

Parallel Pro will soon be shipping from Applied Engineering. Parallel Pro is an intelligent parallel printer interface card designed for the Apple II Plus, IIC and IIGs. Powerful features like Apple IIGs super high resolution graphics, mixed text and graphics, as well as screen dumps are fully supported by Parallel Pro. An optional 8K to 256K buffer card that plugs directly into Parallel Pro will also be available. The optional buffer card can accept standard 64K or 256K chips in any combination up to 256K total. In fact, the 64K or 256K chips can be recognized by the optional buffer card individually or in a set of 8. (256K of buffer memory is equivalent to approximately 250 typewritten pages of text.)

AE's 16K Pocket Rocket is available to Apple II Plus owners to expand a 48K machine to 64K of memory. The original 16K card had a chip count of 22. The new Pocket Rocket (it fits in a shirt pocket) has a chip count of only 4. The 16K Pocket Rocket has the Applied Engineering no hassle 5 year warranty. Applied Engineering continues to provide new product support and innovative developments for Apple II Plus computers.

Applied Engineering is shipping the new 2 MEG Plus for expanding RamWorks II and RamWorks III boards up to 3 Megabytes of memory. The 2 MEG Plus piggyback uses standard 1 MEG DRAMS and is expandable to 2 MEG. With 1 MEG installed, the 2 MEG Plus retails for \$499.00. Fully populated with 2 MEG, it sells for \$799.00.

Applied Engineering is seeking an experienced 6502 machine language programmer to develop software and firmware for its expanding line of Apple peripherals. Minimum 2 years of Apple programming experience required. Send your resume to Applied Engineering, P.O. Box 798, Carrollton, TX 75006, Attention Personnel.

Æ APPLIED ENGINEERING™
The Apple enhancement experts

More Than Just Saving the Registers.....Bob Sander-Cederlof

When you are writing a program it is frequently advantageous to code some parts as subroutines. Some of these subroutines will be useful enough that you may wish to use them in lots of programs. In order to make them usable in a variety of circumstances, you have to be sure that they do not have any hidden side-effects. A subroutine for general use should not change any registers, any memory, or any part of the machine state unless the change is a desired function of that subroutine.

It is getting harder and harder to live up to that ideal, as the Apple II grows. In the beginning there was only the plain vanilla 6502, and at most 48K RAM. There was plenty of memory, plenty of stack space, plenty of zero page. Every subroutine could have its own private work area, and not worry too much about RAM or register usage. Machine state was pretty much the same all day long, with the exception of the text and graphics modes. But now we have the IIgs and the 65816 with more registers, megabytes of RAM to manage, bank switching, shadowing, interrupts, variable speed processing, emulation, and so on endlessly.

It used to be enough to just save and restore the A-, X-, and Y-registers. For example, you might find the following at the beginning and end of a 6502 subroutine:

Beginning		End	
-----		-----	
PHA	Save A	PLA	Restore Y
TXA		TAY	
PHA	Save X	PLA	Restore X
TYA		TAX	
PHA	Save Y	PLA	Restore A
		RTS	

This code saves and restores all three machine registers, so that they may all be used for any other purpose inside the subroutine.

On the other hand, you might not find this code, because most subroutines written in the years gone by used one or more of the three registers to pass parameters. To use such a subroutine, you would first place the values used by the subroutine into the registers; then you would call the subroutine. If the subroutine was supposed to return an answer, it might be returned in the same registers. If the subroutine needed to use page zero at all, it might have its own private locations.

When page zero became over-crowded, we started saving and restoring chunks of page zero. If only a few bytes were needed, they could be saved on the stack. If a larger block was needed, the subroutine might save the block in a private area of RAM outside of page zero.

The IIgs and 65816 are so complicated now that I almost despair

of ever writing a completely safe subroutine. Each subroutine must be carefully analyzed to see what information needs to be saved and restored and what internal states, modes, and values need to be used.

Rather than try to give you a complete entry-setup-exit template, I am going to show you the components and let you construct your own as you require them.

The first item that occurs to me has to do with the native and emulation modes. If you are writing a subroutine that can be called from both modes yet internally must use one particular mode, you need to save the caller's mode. The standard way of doing this is:

Beginning	End
-----	-----
PHP Save m and x	PLP Get caller's mode
SEC (or CLC)	XCE
XCE Get into my mode	PLP Get caller's m,x
PHP Save caller's mode	RTS

Seeing that RTS at the end reminds me of another difficulty. A subroutine cannot possibly know whether it was called with a JSR or a JSL instruction. JSR pushes a two-byte return address on the stack, while JSL pushes a three byte return address. If a subroutine was called with a JSR, it must end with an RTS. Likewise, if the call was made with JSL, the return must be an RTL. You have to decide what you are going to use and write the subroutine accordingly. If you need both, you will have to write two different subroutines (although they might share some code internally).

If you need to save and restore the three main registers (A, X, and Y) without knowing ahead of time which mode the caller was in, you need to be sure and save all 16-bits of each of them. Here is how I do that:

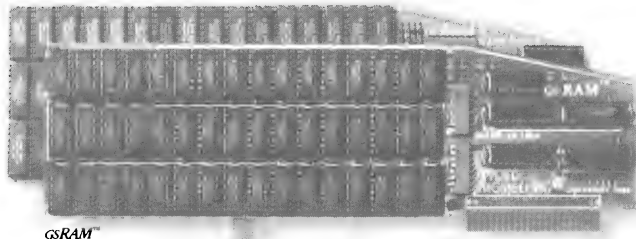
Beginning	End
-----	-----
PHP Save m and x	CLC (Unless still in
CLC Native mode	XCE the native mode)
XCE	REP #\$30
PHP Save caller's mode	PLY Restore Y
REP #\$30 Full 16-bit	PLX Restore X
PHA Save A	PLA Restore A
PHX Save X	PLP
PHY Save Y	XCE Restore caller's mode
	PLP Caller's m and x
	RTS

If the subroutine is being called with a JSL, it is possibly being called from a different bank. That means I may need to worry about the Data Bank Register. In the 65816 there is an 8-bit B-register which contains the bank number to use with all 16-bit addressing modes used to read or write data. If my subroutine needs to use local data or variables which are in the same bank as my subroutine, and the caller was in a

Insist on GSRAM™ When You Buy Your IIGs™

*Expand the IIGs RAM and ROM with the GSRAM or GSRAM Plus
with ROM Pak. Available now with 256K to 8 MEG!*

GSRAM Plus™



GSRAM™

Remember the 16K cards for the II+ and the 64K cards for the IIe? At the time, that much memory seemed like a lot. But when the owners of these memory cards came to us for more memory, many had to throw away their smaller Apple memory cards or try to sell them. Most of our customers told us that had they known about Applied Engineering's larger memory cards when they bought their Apple, they would have purchased them at the same time.

GSRAM and GSRAM Plus are available now, allowing up to 8 MEG of memory expansion. That's 8 times the memory capacity of Apple's card and just look at the benefits that only GSRAM and GSRAM Plus have over Apple's card:

- Lower cost
- Has 6 RAM banks (Apple's card has 4)
- Has memory expansion port
- Has ROM expansion port
- No configuration blocks to set
- No soldered in RAM chips
- Expandable to 8 MEG
- Expands AppleWorks internal limits
- Built-in Hi-Res self-diagnostic software
- 5 year hassle free warranty (Apple has a 90 day warranty)
- Made in USA

GSRAM for More AppleWorks Power

Only GSRAM and GSRAM Plus eliminate AppleWorks internal memory limits, increasing the maximum number of records available from 6,000 to over 25,000 and only GSRAM and GSRAM Plus increase the number of lines permitted in the word processing mode from 6,000 to over 15,000. And only GSRAM and GSRAM Plus offer a built-in printer buffer so you can continue using AppleWorks while your printer is printing. GSRAM and GSRAM Plus even expand the number of lines in the clipboard from 255 to 2047 and will auto segment large files so they can be saved on two or more disks. You can

even have Pinpoint or MacroWorks and your favorite spelling checker in RAM for instant response. GSRAM and GSRAM Plus will even display the time and date right on the AppleWorks screen. Nothing comes close to enhancing AppleWorks so much.

Turn Your IIGs into a Giant

Simply plug GSRAM into the IIGs memory expansion slot and you've got up to 8 megabytes of RAM at your fingertips—all of it instantly and automatically recognized by the IIGs. GSRAM is compatible with all IIGs software, including AppleWorks, as well as BASIC®, ProDOS, DOS 3.3, PASCAL®, "C" and CP/M®.

Grow by Bytes or Megabytes

We offer GSRAM in two configurations so you can increase your memory 256K at a time (GSRAM) or a megabyte at a time (GSRAM Plus). Both offer full compatibility, lower cost than other boards, and easy expandability. And both are extremely low in power consumption. A fully expanded GSRAM operates at only 375 ma, and GSRAM Plus at only 270 ma (even with 6 megabytes on board!)

GSRAM—for Normal Memory Requirements

GSRAM is available with 256K, 512K, 1 MEG or 1.5 MEG of memory already on board. If you don't need the full 1.5 MEG now, you can choose a GSRAM with less memory and expand it up to 1.5 MEG in the future—or upgrade to GSRAM Plus for a small charge.



Steve Wozniak, the creator
of Apple Computer

"In quality,
performance,
compatibility,
expandability and
support, Applied
Engineering's GSRAM
and GSRAM Plus are
number one."

With an optional piggyback card, you can expand GSRAM even higher than 1.5 MEG! (Other cards are only expandable to 1 MEG.)

GSRAM Plus—for Growing by Leaps and Bounds

GSRAM Plus is the first Apple memory card to use 1 MEG RAM chips on the main board. It's available with 1 to 6 MEG on board. If you don't need the whole 6 MEG now, you can buy a GSRAM Plus with less memory and easily expand it in the future.

GSRAM Plus can be expanded up to 8 MEG with an optional piggyback card.

Easy Expansion

Both GSRAM and GSRAM Plus use standard RAM chips that are readily available and just plug right in. So unlike other cards, you'll find expanding your GSRAM or GSRAM Plus easy, convenient and very economical. And with our optional ROM expansion module you can even increase the IIGs's ROM space and all in just one slot.

Quality and Support of the Industry Leader

Applied Engineering is the largest supplier of Apple peripherals in the world. We invented the first large RAM cards for the Apple. With a 5-year "no-hassle" warranty and outstanding technical support, you can be sure GSRAM and GSRAM Plus will deliver the performance you're looking for—or return them within 15 days for a full refund.

GSRAM with 256K	\$169
GSRAM with 512K	\$219
GSRAM with 1 MEG	\$299
GSRAM with 1.5 MEG	\$379
GSRAM with 2-8 MEG	CALL
GSRAM PLUS with 1 MEG	\$459
GSRAM PLUS with 2 MEG	\$759
GSRAM PLUS with 3-8 MEG	CALL

Order today!

See your dealer or call Applied Engineering today, 9 a.m. to 11 p.m. 7 days. Or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 6 1/2% sales tax. Add \$10.00 outside U.S.A.

AE APPLIED ENGINEERING™

The Apple enhancement experts.

(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

CP/M is a registered trademark of Digital Research, Inc.

different bank, I will most likely need to save his B-register and set up my own. There is also a Program Bank Register, called the K-register, which is changed automatically by the JSL and RTL instructions. To save the B-register and make it the same as my K-register, so I can refer to local data and variables, I need the following lines near the beginning of my subroutine:

```
PHB      Save the caller's B-register
PHK      Push Program Bank
PLB      Use it as Data Bank also
```

Then near the end of the subroutine I need the corresponding line:

```
PLB      Restore the caller's B-register
```

It is conceivable that the caller may have the Stack Pointer set to some weird value. In the 6502 the S-register is only 8-bits wide, but in the 65816 it has 16 bits. The stack can be made to start anywhere in bank 0 when you are in native mode. If you have programs that use more than one stack area, you may need to save and restore the caller's stack pointer and use your own inside your subroutine. In particular, if you are going to use Emulation Mode, or call monitor subroutines which do so, you will need to use the standard stack in page 1 inside your subroutine. The following four lines will save the current stack pointer and set up your own, assuming you are already in Native mode with both m- and x-bits cleared for full 16-bit operation:

```
TSC      current stack pointer to A (16-bits)
LDX ##xxxx new value of stack pointer
TXS
PHA      save previous value on new stack
```

You may think of other ways to do this, but something of the sort must be done. If you use the above, then near the end of your subroutine you need to restore the original stack pointer as follows:

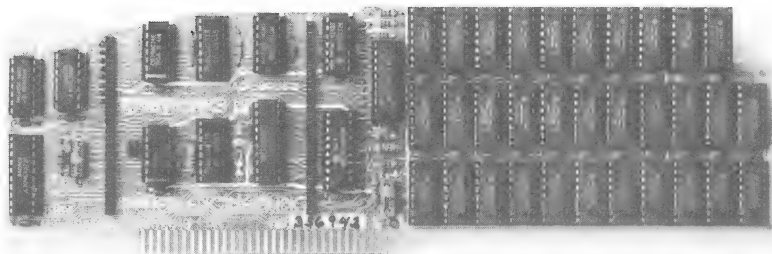
```
PLX
TXS
```

In a 6502 there is only one page zero, at addresses \$0000 through \$00FF. In the 65816 page zero may start anywhere in bank 0, because there is a 16-bit D-register which defines the starting address of the so-called "direct" page. If your programs fiddle with the D-register, then your subroutine might need to save and restore it. If you are calling any of the old monitor subroutines, be aware that they require a "standard" page zero. In 16-bit mode you can do it this way:

Save	Restore
-----	-----
PHD Old D-reg	PLD Get old D-reg
PEA \$xxxx	
PLD New D-reg	

RamWorks® III

Patented Performance from the Recognized Leader



With battery backed RAM port, RGB port, increased memory capacity, full software compatibility and more compact design, RamWorks III is a generation ahead.

RamWorks III is the newest 3rd generation RAM card for the Apple IIe. It incorporates all of the technology and improvements that years of experience and over a hundred thousand sales have given us. By selling more memory cards than anyone else and listening to our customers, we were able to design a memory card that has the ultimate in performance, quality, compatibility and ease of use. A design so advanced it's patented. We call it RamWorks III, you'll call it awesome!

The AppleWorks Amplifier.

While RamWorks III is recognized by all memory intensive programs, NO other expansion card comes close to offering the multitude of enhancements to AppleWorks that RamWorks III does. Naturally, you'd expect RamWorks III to expand the available desktop, after all Applied Engineering was a year ahead of everyone else *including Apple* in offering more than 55K, and we still provide the largest AppleWorks desktops available. But a larger desktop is just part of the story. Look at all the AppleWorks enhancements that even Apple's own card does not provide and *only* RamWorks III does. With a 256K or larger RamWorks III, *all* of AppleWorks (including printer routines) will automatically load itself into RAM dramatically increasing speed by eliminating the time required to access the program disk drive. Switch from word processing to spreadsheet to database at the speed of light with no wear on disk drives.

Only RamWorks eliminates AppleWorks' internal memory limits increasing the maximum number of records available from 1,350 to over 25,000. *Only* RamWorks increases the number of lines permitted in the word processing mode from 2,250 to over 15,000. And *only* RamWorks offers a built-in printer buffer, so you won't have to wait for your printer to stop before returning to AppleWorks. RamWorks even expands the clipboard. And auto segments large files so they can be saved on two or more disks. You can even have Pinpoint or MacroWorks and your favorite spelling checker in RAM for instant response.

RamWorks, *nothing* comes close to enhancing AppleWorks so much.

The Most Friendly, Most Compatible Card Available.

Using RamWorks III couldn't be easier because it's compatible with more off-the-shelf software than any other RAM card. Popular programs like AppleWorks, Pinpoint, Catalyst, MouseDesk, HowardSoft, FlashCalc, Pro-File, Managing Your Money, SuperCalc 3a, and MagiCalc to name a few (and *all* hardware add on's like ProFile and Sider hard disks). RamWorks is even compatible with software written for Apple cards. But unlike other cards, RamWorks plugs into the IIe auxiliary slot providing our super sharp 80 column text (U.S. Patent #4601018) in a completely integrated system while leaving expansion slots 1 through 7 available for other peripheral cards.

RamWorks III is compatible with all

Apple IIe's, enhanced, unenhanced, American or European versions.

Highest Memory Expansion.

Applied Engineering has always offered the largest memory for the IIe and RamWorks III continues that tradition by expanding to 1 full MEG on the main card using standard RAMs, more than most will ever need (1 meg is about 500 pages of text)...but if you do ever need more than 1 MEG, RamWorks III has the widest selection of expander cards available. Additional 512K, 2 MEG, or 16 MEG cards just snap directly onto RamWorks III by plugging into the industry's only low profile (no slot 1 interference) fully decoded memory expansion connector. You can also choose non-volatile, power independent expanders allowing permanent storage for up to 20 years.

It Even Corrects Mistakes.

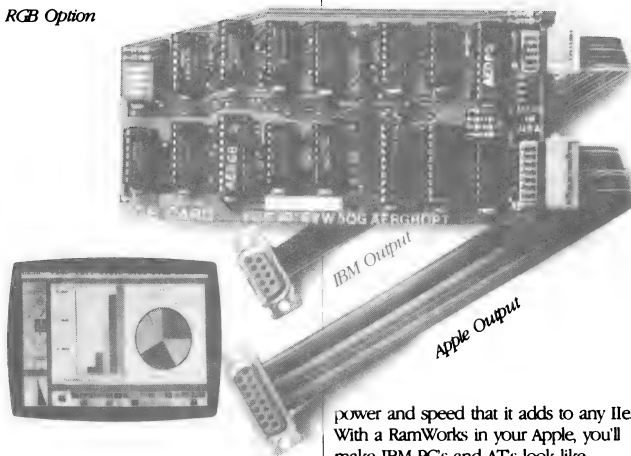
If you've got some other RAM card that's not being recognized by your programs, and you want RamWorks III, you're in luck. Because all you have to do is plug the memory chips from your current card into the expansion sockets on RamWorks to recapture most of your investment!

The Ultimate in RGB Color.

RGB color is an option on RamWorks and with good reason. Some others combine RGB color output with their memory cards, but that's unfair for those who don't need RGB *and* for those that do. Because if you don't need RGB

Applied Engineering doesn't make you buy it, but if you want RGB output you're in for a nice surprise because the RamWorks RGB option offers better color graphics plus a more readable 80 column text (that blows away any composite color monitor). For only \$129 it can be added to RamWorks giving you a razor sharp, vivid brilliance that most claim is the best they have ever seen. You'll also appreciate the multiple text colors (others only have green) that come standard. But the RamWorks RGB option is more than just the ultimate in color output because unlike others, it's fully compatible with all the Apple standards for RGB output control, making it more compatible with off-the-shelf software. With its FCC certified design, you can use almost any RGB monitor because only the new RamWorks RGB option provides both Apple standard and IBM standard RGB outputs (cables included). The RGB option plugs into the back of RamWorks with no slot 1 inter-

RGB Option



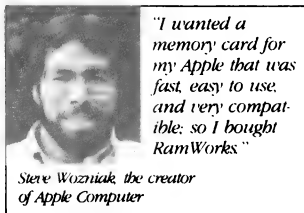
ference and remember you can order the RGB option with your RamWorks or add it on at a later date.

True 65C816 16 Bit Power.

RamWorks III has a built-in 65C816 CPU port for direct connection to our optional 65C816 card. The only one capable of linearly addressing more than 1 meg of memory for power applications like running the Lotus 1-2-3™ compatible program, VIP Professional. Our 65C816 card does not use another slot but replaces the 65C02 yet maintains full 8 bit compatibility.

Endorsed by the Experts.

A+ magazine said "Applied Engineering's RamWorks is a boon to those who must use large files with AppleWorks...I like the product so much that I am buying one for my own system." inCider magazine said "RamWorks is the most



Steve Wozniak, the creator of Apple Computer

"I wanted a memory card for my Apple that was fast, easy to use, and very compatible: so I bought RamWorks."

powerful auxiliary slot memory card available for your IIe, and I rate it four stars...For my money, Applied Engineering's RamWorks is king of the hill."

Apple experts everywhere are impressed by RamWorks's expandability, versatility, ease of use, and the sheer

power and speed that it adds to any IIe. With a RamWorks in your Apple, you'll make IBM PC's and AT's look like slowpokes.

Quality and Support of the Industry Leader

RamWorks III is from Applied Engineering, the largest, most well supported manufacturer of Apple peripherals and the inventor of large RAM cards for the Apple. With our 5 year no hassle warranty and outstanding technical support, you're assured of the most trouble free product you can buy.

It's Got It All

- 15 day money back guarantee
- 5 year hassle free warranty

- Built-in super sharp 80 column display. (U.S. Patent #4601018)
- Expandable to 1 MEG on main card
- Expandable to 16 meg with expander cards, with NO slot 1 interference
- Can use 64K or 256K RAMs
- Powerful linear addressing 16 bit coprocessor port
- Automatic AppleWorks expansion up to 3017K desktop
- Accelerates AppleWorks
- Built-in AppleWorks printer buffer
- The only large RAM card that's 100% compatible with all IIe software
- RamDrive™ the ultimate disk emulation software included free
- Memory is easily partitioned allowing many programs in memory at once
- Compatible, RGB option featuring ultra high resolution color graphics and multiple text colors, with cables for both Apple and IBM type monitors
- Built-in self diagnostics software
- Lowest power consumption (U.S. Patent #4601018)
- Takes only one slot (auxiliary) even when fully expanded
- Socketed and user upgradeable
- Software industry standard
- Advanced Computer Aided Design
- Used by Apple Computer, Steve Wozniak and virtually all software companies
- Displays date and time on the AppleWorks screen with any PRO-DOS compatible clock
- Much, much more!

RamWorks III with 64K	\$179
RamWorks III with 256K	\$199
RamWorks III with 512K	\$249
RamWorks III with 1 MEG	\$329
RamWorks III with 1.5 MEG	\$489
RamWorks III with 2 to 16 MEG	CALL
65C816 16 Bit Card	\$159
RGB Option	\$129
Optional Software:	
Pinpoint with RAM	
Enhancement Software	\$79
VIP Professional	\$219

RamWorks III. The industry standard for memory expansion of the Apple IIe. ORDER YOUR RamWorks III TODAY. See your dealer or call (214) 241-6060, 9 a.m. to 11 p.m., 7 days, or send check or money order to Applied Engineering.

MasterCard, Visa and C.O.D. welcome. Texas residents add 6¼% sales tax. Add \$10.00 if outside U.S.A.

AE Applied Engineering
P.O. Box 798, Carrollton, TX 75006
(214) 241-6060

So far I have only been worrying about the 65816 itself. What about all the Apple soft switches? The //e and //c were enough of a challenge, with all the memory banking and 80-column options. Now along comes the IIgs, an order of magnitude more complex!

In fact it is so complex that I have not mastered it yet. For now all I am going to point out is the rules for calling the old monitor subroutines and the new "tools". To use the old monitor subroutines you must be in Emulation mode, and running in bank 0 with normal page zero and stack pointers. The monitor itself must "be" in bank 0. Since the ROM is really in bank FF, you have the option of copying it to RAM in bank 0 or using the soft switch at \$C081, \$C089, \$C082, or \$C08A to map ROM into bank 0.

To call the new "tools" you must be in full 16-bit native mode. I think that is just about the only requirement. Then you use JSL to call the tools via the vectors in bank \$E1.

Beyond this elementary level, it is easy to get lost. It reminds me of the "twisty little passages" in the Original Adventure Game. My advice: be careful; keep close track of everything you change; put everything back the way you found it. If your subroutine works, you may have done it correctly. Maybe not. You might not find out what you left out till next year when you try to use it from a new environment.

Recent articles in Call APPLE have discussed some of the same ideas, and you may profit by reading them. Look at David Sparks' "The Ethical Subroutine" in the June 1987 issue, and Bob Bishop's "Modular Assembly Language Programming" series in earlier issues this year.

Correction to ProDOS Patcher.....Bob Sander-Cederlof

Mike McConnell called with a correction to my Applesoft ProDOS Patcher that affected its ability to find and fix versions 1.1.1 and 1.2. Lines 430 and 440 PEEKed at A, A+1, A+2, and A+3; in fact they should be PEEKing at A, A+3, A+6, and A+9. Change those two lines to:

```
430 IF PEEK(A)<>157 OR PEEK(A+3)<>157 THEN 500
440 IF PEEK(A+6)<>157 OR PEEK(A+9)<>157 THEN 500
```

Then I noticed an error in the REMark at line 450. What it should say is:

```
450 REM ---FOUND VERSION 1.1.1 OR 1.2,
      SO CHANGE "STA" TO "LDA"---
```

Using the S-C Macro Assembler with an Echo Speech Synthesizer
..... Rick Hayner

In 1977, I went to college to train to become a computer programmer. I was programming an IBM 360-30 main frame, with output which was not usable by me, since I am totally blind, and couldn't read the print output. My first programming language was IBM 360 Assembly Language. Depending on sighted readers who knew nothing about computers to read the listings, and having them help me find errors on IBM cards which sometimes had been punched on a keypunch which didn't print on the tops of the cards, was enough to drive a sane person crazy. Ever since then, I have had the desire to become a good Assembly Language programmer.

After getting an Apple //e, in 1985, I thought that using the ECHO Speech Synthesizer with an editor/assembler would be great. The problem was finding one which would talk.

While talking with Larry Skutchan, a friend of mine who is also a blind programmer, I asked him if he knew about an assembler that would talk, and he recommended the S-C Macro Assembler. I work part time for a company on designing software to help physically handicapped people. My boss, Scott Atwood, asked me if I would be willing to learn 6502 Assembly Language, and if I knew of an assembler which would talk, and I told him about the S-C Macro Assembler. About a week later, Scott brought the assembler to me, and using the instructions in the manual, which I was able to obtain as textfiles on diskette, I had the assembler talking within 10 minutes of getting it.

The ECHO Speech Synthesizer uses a program called Textalker, which resides in the language card area. This means I must use the "main memory" version of the S-C Macro Assembler. The instructions for patching the assembler gave the address of the instructions which had to be patched out in the main memory version, so there were no problems encountered in patching the assembler.

The thing which stops most Machine Language programs from talking is a "JSR \$FE89" followed by a "JSR \$FE93". These instructions are like doing an "IN#0" and a "PR#0". The problem is that Textalker changes the I/O hooks, and these routines disconnect Textalker. Changing these two instructions to six "NOP" instructions, allows the S-C Macro Assembler to talk. Another thing which stops programs from talking, is people sometimes store data directly into screen memory locations, instead of using "COUT". Fortunately, the S-C Macro Assembler doesn't do this, but you do have to patch out the JSR's mentioned above.

I really like the S-C Macro Assembler. I haven't figured out how to use all of its features yet, but I use it almost every day, and always learn something new about it every time I use it.

I have written some macros for controlling the Echo, and I have included them here for other Echo users.

I want to thank all the people at S-C Software for their help in learning to use the S-C Macro Assembler. They have always been willing to answer all questions, whether they have been about the assembler or about programming. I would recommend the S-C Macro Assembler to anyone who is interested in Assembly Language, whether they are blind or sighted.

```

1000 *SAVE ECHO MACROS
1010 *-----
1020 *
1030 *   Macros for ECHO Speech Synthesizer, by Rick Hayner
1040 *
1050 *   Requires Textalker Version 3.1.1 or later.
1060 *
1070 *-----
1080 *   Generic ECHO Command: >ECHO "X"  (X is command letter)
1090 *-----
1100 *       .MA ECHO
1110 *       LDA #$85      Control-E
1120 *       JSR MON.COUT
1130 *       LDA #"1"      Command letter
1140 *       JSR MON.COUT
1150 *       .EM
1160 *-----
1170 *   TURN ECHO OFF: >OFF
1180 *       .MA OFF
1190 *       >ECHO "O"
1200 *       .EM
1210 *-----
1220 *   TURN ECHO ON: >ON
1230 *       .MA ON
1240 *       >ECHO "B"
1250 *       .EM
1260 *-----
1270 *   PUT ECHO IN TALK MODE: >TALK
1280 *       .MA TALK
1290 *       >ECHO "T"
1300 *       .EM
1310 *-----
1320 *   PUT ECHO IN LETTER MODE: >LETTER
1330 *       .MA LETTER
1340 *       >ECHO "L"
1350 *       .EM
1360 *-----
1370 *   PUT ECHO IN WORD MODE: >WORD
1380 *       .MA WORD
1390 *       >ECHO "W"
1400 *       .EM
1410 *-----
1420 *   TURN UPPERCASE SPELL ON: >USPELL
1430 *       .MA USPELL
1440 *       >ECHO "K"
1450 *       .EM
1460 *-----
1470 *   TURN UPPERCASE SPELL OFF: >USPELLOFF
1480 *       .MA USPELLOFF
1490 *       >ECHO "N"
1500 *       .EM
1510 *-----
1520 *   PUT ECHO IN SOME PUNCTUATION: >SOME
1530 *       .MA SOME
1540 *       >ECHO "S"
1550 *       .EM
1560 *-----
1570 *   PUT ECHO IN MOST PUNCTUATION: >MOST
1580 *       .MA MOST
1590 *       >ECHO "M"
1600 *       .EM
1610 *-----
1620 *   PUT ECHO IN ALL PUNCTUATION: >ALL
1630 *       .MA ALL
1640 *       >ECHO "A"
1650 *       .EM
1660 *-----

```



SPECIAL !!! EXPANDED RAM/ROM BOARD: \$39.00

Similar to our \$30 RAM/ROM dev board described below. Except this board has two sockets to hold your choice of 2-2K RAM, 2-2K ROM or even 2-4K ROM for a total of 8K. Mix RAM and ROM too. Although Apple limits access to only 2K at a time, soft switches provide convenient socket selection. Hard switches control defaults.

IMPROVED !!! II IN A MAC (ver 2.0): \$75.00

Now includes faster graphics, UniDisk support and more! Bi-directional data transfers are a snap! This Apple II emulator runs DOS 3.3/PRODOS (including 6502 machine language routines) on a 512K MAC or MACPLUS. All Apple II features are supported such as HI/LO-RES graphics, 40/80 column text, language card and joystick. Also included: clock, RAM disk, keyboard buffer, on-screen HELP, access to the desk accessories and support for 4 logical disk drives. Includes 2 MAC diskettes (with emulation, communications and utility software, plus DOS 3.3 and PRODOS system masters, including Applesoft and Integer BASIC) and 1 Apple II diskette.

SCREEN.GEN: \$35.00

Develop HI-RES screens for the Apple II on a Macintosh. Use MACPAINT (or any other application) on the MAC to create your Apple II screen. Then use SCREEN.GEN to transfer directly from the MAC to an Apple II (with SuperSerial card) or IIC. Includes Apple II diskette with transfer software plus fully commented SOURCE code.

MIDI-MAGIC for Apple //c: \$49.00

Compatible with any MIDI equipped music keyboard, synthesizer, organ or piano. Package includes a MIDI-out cable (plugs directly into modem port - no modifications required!) and 6-song demo diskette. Large selection of digitized QRS player-piano music available for 19.00 per diskette (write for catalog). MIDI-MAGIC compatible with Apple II family using Passport MIDI card (or our own input/output card w/drum sync for only \$99.00).

FONT DOWNLOADER & EDITOR: \$39.00

Turn your printer into a custom typesetter. Downloaded characters remain active while printer is powered. Use with any Word Processor program capable of sending ESC and control codes to printer. Switch back and forth easily between standard and custom fonts. Special functions (like expanded, compressed etc.) supported. Includes HIRES screen editor to create custom fonts and special graphics symbols. For Apple II, II+, //e. Specify printer: Apple Imagewriter, Apple Dot Matrix, C.Itoh 8510A (Prowriter), Epson FX 80/85, or Okidata 92/192.

* **FONT LIBRARY DISKETTE #1: \$19.00** contains lots of user-contributed fonts for all printers supported by the Font Downloader & Editor. Specify printer with order.

DISASM 2.2e : \$30.00 (\$50.00 with SOURCE Code)

Use this intelligent disassembler to investigate the inner workings of Apple II machine language programs. DISASM converts machine code into meaningful, symbolic source compatible with S-C, LISA, ToolKit and other assemblers. Handles data tables, displaced object code & even provides label substitution. Address-based triple cross reference generator included. DISASM is an invaluable machine language learning aid to both novice & expert alike. Don Lancaster says DISASM is "absolutely essential" in his ASSEMBLY COOKBOOK.

The 'PERFORMER' CARD: \$39.00 (\$59.00 with SOURCE Code)

Converts a 'dumb' parallel printer I/F card into a 'smart' one. Simple command menu. Features include perforation skip, auto page numbering with date & title, large HIRES graphics & text screen dumps. Specify printer: MX-80 with Grafrax-80, MX-100, MX-80/100 with Grafraxplus, NEC 8092A, C.Itoh 8510 (Prowriter), OkiData 82A/83A with Okigraph & OkiData 92/93.

'MIRROR' ROM: \$25.00 (\$45.00 with SOURCE Code)

Communications ROM plugs directly into Novation's Apple-Cat Modem card. Basic modes: Dumb Terminal, Remote Console & Programmable Modem. Features include: selectable pulse or tone dialing, true dialtone detection, audible ring detect, ring-back, printer buffer, 80 col card & shift key mod support.

RAM/ROM DEVELOPMENT BOARD: \$30.00

Plugs into any Apple slot. Holds one user-supplied 2Kx8 memory chip (6116 type RAM for program development or 2715 EPROM to keep your favorite routines on-line). Maps into \$Cn00-CnFF and \$C800-CFFF.

C-PRINT For The APPLE //c: \$69.00

Connect standard parallel printers to an Apple //c serial port. Separate P/S included. Just plug in and print!

Unless otherwise specified, all Apple II diskettes are standard (not copy protected!) 3.3 DOS.

Avoid a \$3.00 handling charge by enclosing full payment with order. VISA/MC and COD phone orders OK.

RAK-WARE 41 Ralph Road W. Orange N.J 07052 (201) 325-1885



From the ProDOS version of the S-C Macro Assembler, how does one direct the assembly listing to a text file for later inclusion in a word processor file? Articles in AAL some years ago showed how to do this for the DOS 3.3 version, but the techniques given do not work with the ProDOS version.

After some head scratching and some snooping, I did find a way that works. By patching the PBITS entries for the OPEN and WRITE commands I can make them into DIRECT commands. (Without the patches, OPEN and WRITE are not legal as DIRECT commands.) I'll show how to make these two patches in a moment.

Once the patches are installed, by merely typing "OPEN LISTING" and "WRITE LISTING" an output textfile starts absorbing all screen-directed information. I make sure I am ready to assemble, and then type the OPEN and WRITE commands. At this point the screen output disappears, because it is all going to the text file instead. Then I type the ASM command, and the listing is written on the text file. At the end of the assembly the text file is automatically closed, and the prompt re-appears.

There are some "gotchas". First, the S-C Macro Assembler has a fixed number of file buffers. If I use one of them to send the listing to a text file, it is not possible to also have a target file (.TF file) at the same time. This could restrict your possibilities when working with large programs, because the object code does have to go somewhere. Judicious use of the .TA, .DU, or .PH directives could enable you to assemble even large programs while directing the listing to a text file.

Second, the text file will not be truncated if a short listing is written on a previously existing file which was longer. The solution is to always DELETE such a file before you OPEN it.

As a test, I assembled the S-C DisAssembler (an excellent product, by the way) and sent the listing to a text file. I first removed the .TI line so that the resulting file would not have page breaks. Working with two floppy drives, the source code on one and the output listing file on the other, it produced a 177 block text file in just under five minutes. I moved the file into Appleworks with no problems. After adjusting the margins to 0 inches it was 123K, 2298 lines, and saved onto a 95K AWP file.

Now, how do you make those two patches? First you have to find the two bytes to change. Bob has produced several slightly different versions so this requires a little snooping. With the S-C Macro Assembler loaded and running, enter the monitor command "\$B900.B99F". This will display the area containing the two bytes that need changing. In some of the monitor versions it will display in both hex and ASCII, but in most just in hex. There are two strings of bytes you need to find:

```
4F 50 45 CE xx xx 2D
57 52 49 54 C5 xx xx 21
```

The first string starts somewhere near \$B918, and the second somewhere near \$B958. The "xx xx" indicates two bytes which could have various values. The "2D" and "21" are the two bytes to patch. In my copy of the S-C Macro Assembler, these were at \$B91F and \$B95F, respectively. Be sure you find and use the correct addresses for yours. Change the 2D to 0D, and the 21 to 01, as follows:

```
: $B91F:0D N B95F:01
```

That is all there is to it! Note that the change is temporary, because it does not modify the SCASM.SYSTEM file. You could find the appropriate place and do that if you wish, but it is just as effective to create a small EXEC file to install the changes whenever you need them.

EnterSoft

Basic-like Macros for the S-C Macro Assembler. These macros are Integer Basic Macros which can perform most of the standard Basic commands using 8/16/32/64 bit manipulation. Included are a set of graphic commands to perform block graphics and other Hi-Res Animation. Available in either DOS 3.3 or ProDos Format. Either one for \$30.00 or BOTH for \$50.00.

A Shape Table Program:

A shape table program unlike any other program you have ever run. This one is completely set up in a logical manner. All disk operations are under one section. All editing functions are contained in one place. All Hi-Res functions are in one place. The entire source code is included to allow you to modify it as you need to! The DOS 3.3 Version is only \$20.00 - AND SO IS THE PRODOS VERSION!!!! Both Versions for only \$30.00. ORDER YOURS TODAY!!!!

To Order:

Send check or Money Order To:

Mark Manning
c/o Simulacron 1/Baggy Games
P.O. Box 591894
Houston, Tx 77259-1894

Upgrades for EnterSoft from DOS 3.3 to ProDos is \$20.00. Get yours today!

Thanks Everyone For Your
Comments and Support!

Printing a Tiny Address Booklet.....Bob Sander-Cederlof

Do you have trouble remembering all the phone numbers and addresses of your family and friends? I sure do. At work I keep a Rolodex file for the most frequently needed ones, but I can hardly carry it around in my pocket. I could buy a little blank address book at any store, but my handwriting is too poor and they all take up too much space.

I had a vision of a tiny address booklet, no bigger than a credit card, which could hold about 100 names, addresses, and phone numbers. I could carry it in my credit card wallet, so it would almost always be with me when I need it.

It is easy enough to use a word processor to type all the information into my Apple. Last year I did just that, and printed it all out in the condensed mode on my NEC 8023 dot matrix printer. I then cut the printout into little pages and pasted them up in booklet format. Using a Minolta copying machine, I "xeroxed" the paste-up onto both sides of the same sheet of paper. Then using scissors and staples I assembled a little booklet the same size as a credit card, with 12 pages counting the front and back covers. I left those covers plain white, and covered them with clear packing tape, so it could last a little longer.

Indeed it did last, for about a year. Now there are so many changes that it is getting hard to read. I need to bring it up to date. But can the computer do a little more of the work?

Of course it can. Especially with the help of the new Epson EX-800 dot matrix printer I just bought, for only \$399 plus tax at the local Soft Warehouse. It prints a lot faster than my other printers, which is why I bought it. But it also has a lot of new features. One of them is the ability to print subscripts and superscripts. I found out that in this mode, with Condensed and Elite also selected, the printout is not only very small, but very clear and easy to read. The characters are shorter, allowing me to adjust the vertical spacing to 5/72 inch per line. This gets over 150 lines on each sheet of paper! It also lets me get 50 lines (including top and bottom margin) in the 3.5 inches of my booklet pages. And I can print 33 characters per line on each one of the pages, leaving a nice left and right margin. The Epson EX-800 makes two passes over each line in this mode, so that the miniature print is "near letter quality".

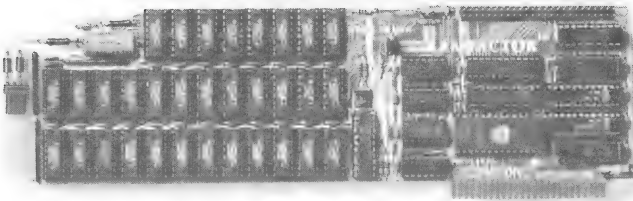
The printer helps, but I still wanted some automatic way to layout the individual pages and print them in booklet order, on BOTH SIDES of the same sheet of paper. I started to do it in Applesoft, but suddenly it seemed easier in Assembly Language. The resulting program prints all of the "front sides" and then waits while I take the paper out of the printer, turn it over, and put it back in. Then it prints the "back sides". All I have to do is fold the sheet down the middle, slice it into three 4-page folders, stack them up, and staple them together. It worked beautifully, and the result is nestling nicely in my shirt pocket right now.

RamFactor™

Boot any II+, IIe, or IIGS and be up and running your favorite software in less than 1 second.

Now anyone with an Apple II+, Franklin, Laser 128, Apple IIe or IIx preferring to use slots 1 through 7 can now enjoy the speed and performance that they've been waiting for.

With RamFactor, you'll be able to instantly add another 256K, 512K, or a full 1 MEG on the main board and an expansion connector can address an additional 16 MEG of memory for possible future expansion. Since virtually all software is automatically compatible with RamFactor, you'll immediately be able to load programs into RamFactor for instantaneous access to information. You'll also be able to store more data for larger word processing documents, bigger data bases, and expanded spreadsheets. And with the battery back-up option, you can switch on your Apple and your favorite software will be up and running in less than 1 second!



Very Compatible

All the leading software is already compatible with RamFactor. Programs like AppleWorks, Pinpoint, BPI, Managing Your Money, Dollars and Sense, SuperCalc 3A, PFS, MouseWrite, MouseDesk, MouseCalc, Sensible Speller, Applewriter IIe, Business Works, ReportWorks, Catalyst 3.0 and more. And RamFactor is fully ProDos, DOS 3.3, Pascal 1.3 and CP/M compatible. In fact, no other slot 1-7 memory card is more compatible with commercial software.

AppleWorks Power

There are other slot 1-7 cards that give AppleWorks a larger desktop, but that's the end of their story. But RamFactor is the only slot 1-7 card that increases AppleWorks internal memory limits, increasing the maximum number of records in the database and lines permitted in the word processor, and RamFactor is the only standard slot card that will automatically load all of AppleWorks into RAM dramatically increasing speed and eliminating the time required to access the program disk, it will even display the time and date on the AppleWorks screen with any ProDos clock. RamFactor will automatically segment large files so they can be saved on 5¼", 3½", and hard disks. All this performance is available on the Apple II+,

IIe, Franklin, or Laser 128 when used with an 80 column display. RamFactor, no other standard slot card comes close to enhancing AppleWorks so much.

The "Instant On" Apple.



With the RamCharger battery backup option, RamFactor will retain programs and data during power failures or even when you turn your computer off. And because RamFactor plugs into slots 1-7, it's the only memory card that can cold boot an Apple without using slow floppy or hard disks. An available "Y" cable allows one RamCharger to power two RamFactors.

Powerful Program Switcher

With RamFactor, you can organize memory into multiple work areas and switch between them. Each work area can contain different programs and even different operating systems. Now you can switch from one program to another or even switch from AppleWorks to DOS 3.3 to CP/M to Pascal to ProDos in under a second. And with our battery back-up option, you can have permanent storage for up to 20 years.

Features:

- 256K to 1 MEG on main board. Expansion connector for possible further expansion to 16 MEG.
- Fully Apple II Memory Expansion compatible
- Compatible with Apple IIe, II+, IIx, Franklin and Laser 128
- Battery backup option allows you to turn on your Apple and run your favorite programs in less than 1 second!
- Automatically recognized by ProDos, DOS 3.3, Pascal, and standard CP/M
- Built-in RamDrive™ software (a true RAM disk not disk caching)
- Systems are directly bootable from RamFactor if desired
- Built-in self diagnostic software
- Automatic expansion with AppleWorks 1.3 or 2.0
- Allows Apple II+ and IIe to run your AppleWorks without buying any additional software
- Accelerates AppleWorks
- Displays time and date on the AppleWorks screen with any ProDos clock
- Fits any I/O slot except slot 3
- Fully socketed and user upgradeable
- Much, much more

RamFactor with 256K	\$239
RamFactor with 512K	\$279
RamFactor with 1 MEG	\$359
RamFactor with 2-16 MEG	CALL
RamCharger backup option	\$179
"Y" cable	\$24
<small>(Allows one RamCharger to power two RamFactors.)</small>	

Order RamFactor today... with 15 day money back guarantee and our "no hassle" five year warranty. See your dealer or call (214) 241-6060, 9 a.m. to 11 p.m., 7 days, or send check or money order to Applied Engineering, MasterCard, Visa and C.O.D. welcome. Texas residents add 6% sales tax. Add \$10.00 if outside U.S.A.

AE Applied Engineering™
The Apple enhancement experts
(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

To understand my program a little more easily, it might help to take three small pieces of paper, stack them up, fold the stack in half, and call it a booklet. Then label the pages from front to back. Label the front cover "FC", the inside pages 0 through 9, and the back cover "BC". Then take the booklet apart and look at each sheet. I call the outer sheet "FC/0--9/BC", the next sheet "1/2--7/8", and the inner sheet "3/4--5/6". My program prints on pages 0 through 9, and leaves FC and BC blank.

My program works under the assumption that the text file containing all the names and addresses is already in memory. It just happens in my case to begin at \$2121. Each character is in ASCII with the high bit on, and the file image ends with a \$00 byte. You could make obvious adjustments to the program to work with text in some other position or a different format.

My program also assumes that there are blank lines between each entry, and no blank lines within any entry. This helps me to easily be sure that an entry is not printed partly on one page and partly on another. I counted the number of lines in my entire directory, finding 341 lines. Since I am printing on ten pages, that is about 34 lines per page. I decided to print at least 32 lines on each page, and then finish printing whatever entry was in progress. This resulted in the last page being a little shorter than the rest, and left enough bottom margin on most pages for adding extra names later.

My program begins by setting up the Epson printer. You will need to modify this setup to correspond with the capabilities of your own printer and interface card. Lines 1220-1250 install the printer hook. As you can see, my printer interface card was in slot 5, which is highly unusual. Yours is probably in slot 1, so you will want to change the "\$C500" to "\$C100". The setup string is shown in lines 1640-1720. The first two lines are control-I commands to the interface card, and the rest are ESCAPE commands to the Epson printer.

The next major activity in my program involves building a table of addresses which point to the beginning of each booklet page in the text buffer. The subroutine which does this is shown in lines 2450-2700. It simply starts with the first page at the given address of \$2121 (probably different in your case), scans to the beginning of the next page and saves the address, and so on for 11 pages. The extra page does not really exist, but it helps me set up another table which is used to mark the END of each page. I store the addresses in two different tables: PAGES and SPAGES. PAGES is used for the beginning and current address in each page, and is modified during printout. SPAGES is used to find the end of each page, and is not modified during printout.

Another value you may want to change is the number 32 in line 2780. This is the minimum number of lines to be printed on each page. It turns out my longest page had 35 lines printed on it. If you use a different booklet size or a different vertical spacing, you will probably want a different value here.

Looking back at the main program, lines 1360-1450 print the front side of the sheet. Line 1470 pauses the program so that I can turn the paper over and re-insert it into the printer, and then lines 1480-1540 print the back side. My subroutine PRINT.TWO.PAGES does the real work here. The page number (0-9) of the left page is in the A-register and that of the right page in the X-register.

PRINT.TWO.PAGES assumes there are a total of 50 lines per page. You should change the number on line 1830 for a different size page. This count includes the top and bottom margin. I forgot to include a little extra for trimming, but it worked out all right for me. The number 42 on line 1860 controls the starting column for the right-hand page. I had a 33-character line, a 9-character center page gap, and the right page starts in column 43. You may need to change this number also.

The number 35 on line 1880 was arrived at experimentally. Between adjusting the tractors left and right to position the paper, and changing this number 35 up or down, you can center the blank space between the left and right pages exactly over the center of the printer paper. This is important, because when we turn the paper over and print on the back side we want the pages to line up correctly.

Lines 1900-1960 print a line for the left-hand page and tab to the starting column for the right-hand page. Then lines 1980-2040 print a corresponding line for the right-hand page. Of course, either or both of these may be blank lines.

I wish I could show you what the finished product looks like, but that is not practical. If you stop by some time, I will show it to you. Meanwhile, you could adapt this to your own printer and needs and make your own booklet.

A project comes to mind, which one of you might like to tackle. How about parameterizing the variables so that it could make booklets of various dimensions and page counts, and integrate it into a word processor? The S-C Word Processor naturally comes to mind, since so many of you have purchased it with the source code.

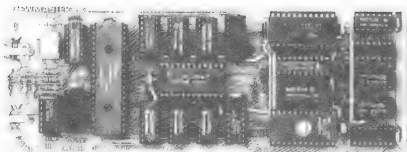
```

0000-      1000  *SAVE PAGINATOR
0018-      1010  *-----
0019-      1020      .DUMMY
001E-      1030      .OR 0          DEFINE PAGE ZERO VARIABLES
001C-      1040  PAGES  .BS 24      Room for 12 pointers
001D-      1050  PAGE   .BS 1
001E-      1060  PNTR   .BS 2
001C-      1070  LPAGE   .BS 1
001D-      1080  RPAGE   .BS 1
001E-      1090  LINES   .BS 1
001E-      1100  TABBER  .BS 1
001E-      1110      .ED          END OF DUMMY SECTION
001E-      1120  *-----
001E-      1130  MON.CH  .EQ $24
001E-      1140  *-----
001E-      1150  BEGIN   .EQ $2121   My text buffer starts here
001E-      1160  *          and ends with a 00 byte.
001E-      1170  *-----
001E-      1180  CROUT   .EQ $FD8E
001E-      1190  COUT    .EQ $FDED
001E-      1200  *-----

```

Viewmaster 80

Now run AppleWorks™ on your II+ *and* keep full 80 column compatibility!



	800K IN SLOT/128K SLOT 128K SLOT 128K	LOW POWER DESIGN	80 COL MN ROM	19.2K BAUD	80 COL MN SLOT 128K	80 COL MN SLOT 128K	80 COL MN SLOT 128K
VIEWMASTER 80	x	x	x	x	x	x	x
SE PRITERM							
WIZARD 80							
VISION 80	x	x		x			
OMNIVISION							
VIEWMAX 80	x			x			
SMARTER	x	x		x	x		
VIDEOTERM							

One look at the chart will give you some of the reasons there's only one smart choice in 80 column cards for your Apple. But the real secret to Viewmaster 80's success is something even better: Total compatibility.

Each Viewmaster 80 includes our powerful AppleWorks expand software, allowing AppleWorks to run on the II+ with only 64K (or more) memory. (We recommend the RamFactor memory card, but any compatible Apple memory card will work.) The software provides our full range of AppleWorks enhancements, including expanded records, word processor, multi-disk saving, time and date display on screen with any PRO-DOS clock, and more!

The Viewmaster 80 works with all 80 column applications, including DOS 3.3, PRO-DOS, CP/M, Pascal, WordStar, Format II, Applewriter II, dBase II, Visicalc, Multiplan, and *hundreds* of others—including AppleWorks.

Here are just a few of the powerful features the Viewmaster 80 delivers for a great price (\$149):

- 80 Characters by 24 lines • Fully compatible with all Apple languages and software • Highest compatibility with existing 80 column software • Very low power consumption • High speed (18 MHz) scroll rate • Upper and lower case characters and true descenders, both inverse and normal; all on-screen editing functions are supported • User-definable cursor shape • Compatible with Apple II, II+ and IIe • Five-year warranty

Call today to order or for more information, 9 a.m. to 11 p.m. seven days, or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 5.5% sales tax. Add \$10.00 outside U.S.A.

CP/M is a registered trademark of Digital Research, Inc.

AE Applied Engineering
P.O. Box 798, Carrollton, TX 75006
(214) 241-6060

```

1210 T
0800- A9 00 1220 LDA #$C500 Hook into my printer card
0802- 85 36 1230 STA $36 in slot 5
0804- A9 C5 1240 LDA /$C500 (Yours is probably in slot 1)
0806- 85 37 1250 STA $37
1260 *---Setup Printer-----
0808- A0 00 1270 LDY #0 Send setup characters to my
080A- B9 4D 08 1280 .1 LDA STRING,Y printer interface card
080D- 20 ED FD 1290 JSR COUT and to my Epson EX-800
0810- C8 1300 INY Printer.
0811- C0 12 1310 CPY #STR.LEN
0813- 90 F5 1320 BCC .1
1330 *---Find the pages-----
0815- 20 D5 08 1340 JSR BUILD.POINTER.TABLE
1350 *---Print the front side-----
0818- 20 8E FD 1360 JSR CROUT
081B- A9 02 1370 LDA #2 Print front side of 1/2--7/8
081D- A2 07 1380 LDX #7
081F- 20 68 08 1390 JSR PRINT.TWO.PAGES
0822- A9 04 1400 LDA #4 Print front side of 3/4--5/6
0824- A2 05 1410 LDX #5
0826- 20 68 08 1420 JSR PRINT.TWO.PAGES
0829- A9 00 1430 LDA #0 Print front side of FC/0--9/BC
082B- A2 09 1440 LDX #9
082D- 20 68 08 1450 JSR PRINT.TWO.PAGES
1460 *---Print the back side-----
0830- 20 5F 08 1470 JSR FLIP.PAPER.OVER
0833- 20 8E FD 1480 JSR CROUT
0836- A9 08 1490 LDA #8 Print back side of 1/2--7/8
0838- A2 01 1500 LDX #1
083A- 20 68 08 1510 JSR PRINT.TWO.PAGES
083D- A9 06 1520 LDA #6 Print back side of 3/4--5/6
083F- A2 03 1530 LDX #3
0841- 20 68 08 1540 JSR PRINT.TWO.PAGES
1550 *---Disconnect Printer-----
0844- A9 00 1560 LDA #$C300 Rehook the 80-column screen
0846- 85 36 1570 STA $36
0848- A9 C3 1580 LDA /$C300
084A- 85 37 1590 STA $37
084C- 60 1600 RTS
1610 *-----
1620 * Printer Setup String
1630 *-----
084D- 09 4E 1640 STRING .HS 09.4E Turn off Video
084F- 09 58 1650 .HS 09.58 Bit 7 = 0
0851- 1B 53 01 1660 .HS 1B.53.01 Subscript
0854- 1B 41 05 1670 .HS 1B.41.05 5/72 inch vertical
0857- 1B 23 1680 .HS 1B.23 Cancel Bit 7
0859- 1B 3D 1690 .HS 1B.3D Bit 7 = 0
085B- 1B 4D 1700 .HS 1B.4D Elite
085D- 1B 0F 1710 .HS 1B.0F Condensed
12- 1720 STR.LEN .EQ *-STRING
1730 *-----
1740 FLIP.PAPER.OVER
085F- AD 00 C0 1750 .1 LDA $C000 Really just wait until I
0862- 10 FB 1760 BPL .1 manually turn it over.
0864- 8D 10 C0 1770 STA $C010 Signal by typing any key.
0867- 60 1780 RTS
1790 *-----
1800 PRINT.TWO.PAGES
0868- 85 1B 1810 STA LPAGE Left Page Number
086A- 86 1C 1820 STX RPAGE Right Page Number
086C- A9 32 1830 LDA #50 Print 50 Lines per Page
086E- 85 1D 1840 STA LINES
1850 *---Print left side-----
0870- A9 2A 1860 .1 LDA #42 Right Page begins in col. 43
0872- 85 1E 1870 STA TABBER
0874- A9 23 1880 LDA #35 Left Margin to center on page
0876- 85 24 1890 STA MON.CH
0878- A5 1B 1900 LDA LPAGE
087A- 20 A1 08 1910 JSR SETUP.PAGE.PNTR
087D- B0 03 1920 BCS .2 ...end of this page
087F- 20 BD 08 1930 JSR PRINT.LINE
0882- 20 CB 08 1940 .2 JSR TAB.TO.MIDDLE
0885- A5 1B 1950 LDA LPAGE
0887- 20 B2 08 1960 JSR UPDATE.PAGE.PNTR

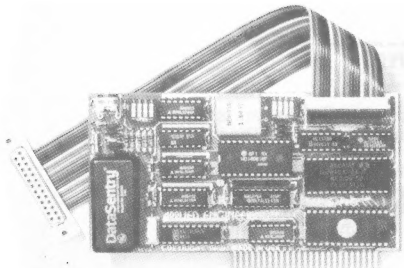
```

```

1970 *---Print right side-----
088A- A5 1C 1980 LDA RPAGE
088C- 20 A1 08 1990 JSR SETUP.PAGE.PNTR
088F- B0 03 2000 BCS .3 ...end of this page
0891- 20 BD 08 2010 JSR PRINT.LINE
0894- 20 8E FD 2020 .3 JSR CROUT
0897- A5 1C 2030 LDA RPAGE
0899- 20 B2 08 2040 JSR UPDATE.PAGE.PNTR
2050 *-----
089C- C6 1D 2060 DEC LINES Count down for the page
089E- D0 D0 2070 BNE .1 More lines on this page
08A0- 60 2080 RTS Finished
2090 *-----
2100 SETUP.PAGE.PNTR
08A1- OA 2110 ASL Double page number for index
08A2- AA 2120 TAX
08A3- B5 00 2130 LDA PAGES,X Get pointer to current
08A5- 85 19 2140 STA PNTR to current place on page
08A7- DD 4A 09 2150 CMP SPAGES+2,X ...compare to end of page
08AA- B5 01 2160 LDA PAGES+1,X
08AC- 85 1A 2170 STA PNTR+1
08AE- FD 4B 09 2180 SBC SPAGES+3,X Leave .CS. if end of page.
08B1- 60 2190 RTS ...or .CC. if not end yet.
2200 *-----
2210 UPDATE.PAGE.PNTR
08B2- OA 2220 ASL Double page number for index
08B3- AA 2230 TAX
08B4- A5 19 2240 LDA PNTR Save current place on page
08B6- 95 00 2250 STA PAGES,X for next line
08B8- A5 1A 2260 LDA PNTR+1
08BA- 95 01 2270 STA PAGES+1,X
08BC- 60 2280 RTS
2290 *-----
2300 PRINT.LINE
08BD- 20 33 09 2310 .1 JSR NEXT.CHAR Print up to <RETURN>
08C0- B0 08 2320 BCS .2 ...or end of text
08C2- C6 1E 2330 DEC TABBER Count the character
08C4- 20 ED FD 2340 JSR COUT and print it.
08C7- 4C BD 08 2350 JMP .1 ...next character
08CA- 60 2360 .2 RTS
2370 *-----
2380 TAB.TO.MIDDLE
08CB- A9 A0 2390 .1 LDA #" " TABBER has remaining
08CD- 20 ED FD 2400 JSR COUT blank count to reach
08D0- C6 1E 2410 DEC TABBER the right side.
08D2- D0 F7 2420 BNE .1
08D4- 60 2430 RTS
2440 *-----
2450 BUILD.POINTER.TABLE
08D5- A9 21 2460 LDA #BEGIN Start at the beginning
08D7- 85 19 2470 STA PNTR of my text buffer.
08D9- 85 00 2480 STA PAGES First page starts here.
08DB- A9 21 2490 LDA /BEGIN Hi byte of same...
08DD- 85 1A 2500 STA PNTR+1
08DF- 85 01 2510 STA PAGES+1
08E1- A9 01 2520 LDA #1 for PAGE = 1 to 11
08E3- 85 18 2530 STA PAGE
2540 *-----
2550 .1 JSR SCAN.TO.END.OF.PAGE
08E8- A5 18 2560 LDA PAGE Save pointer to beginning of page
08EA- OA 2570 ASL double for index
08EB- AA 2580 TAX
08EC- A5 19 2590 LDA PNTR It is also END+1 for
08EE- 95 00 2600 STA PAGES,X previous page.
08F0- 9D 4A 09 2610 STA SPAGES,X
08F3- A5 1A 2620 LDA PNTR+1 Hi-byte of same.
08F5- 95 01 2630 STA PAGES+1,X
08F7- 9D 49 09 2640 STA SPAGES+1,X
2650 *-----
08FA- E6 18 2660 INC PAGE next PAGE
08FC- A5 18 2670 LDA PAGE
08FE- C9 0C 2680 CMP #12 (up to 12)
0900- 90 E3 2690 BCC .1 ...do another page
0902- 60 2700 RTS finished
2710 *-----

```

Serial Pro® The intelligent multifunction card from Applied Engineering.



Serial Pro is a powerful multifunction card for the Apple IIe, II+, IIcs and compatible computers. Serial Pro combines a powerful serial port for interfacing to a printer, modem or other serial device, with a full function clock/calendar. Serial Pro's serial port is compatible with virtually all letter quality and dot matrix printers, including Apple's Imagewriter and Imagewriter II, and can communicate with the fastest external modems at baud rates from 50 to 19,200.

Set up is easy, because unlike other cards, there are no switches to program serial parameters or disks to use. Serial Pro has a built-in setup screen that allows all parameters to be set. And on screen help that makes it super-easy. An on-board battery stores your selections for over 20 years, but you can change them at any time. Serial Pro is the perfect upgrade for Super Serial card owners looking for more performance or a built-in clock.

Serial Port Features

- Fully Super Serial Card compatible but more versatile and easier to use
- Compatible with printers
- Compatible with modems
- Prints graphics to printer (without software) in normal mode, 90° rotation, double size, side by side, and black/white mode
- Variable word length, number of stop bits and parity selection
- Crystal controlled circuits are used to provide ultra accurate baud rates

Clock Calendar Features

- ProDOS, DOS and AppleWorks compatible. Displays time and date on the AppleWorks screen
- Automatically time and date stamps AppleWorks files
- 20 year auto recharging nickel-cadmium battery
- Realtime interrupts
- Built-in on screen time and date settings (no software needed)
- Does not use phantom slots

Serial Pro is perfect for Apple IIe, II+, IIcs or compatible owners in need of a serial port or clock/calendar or both. But unlike other multifunction cards on the market, Serial Pro does not use phantom slots, so all the slots in your computer are useable.

Serial Pro comes complete with manual, cable and instructions for connecting to all the leading printers and modems.

\$159

Order Today! See your dealer or call Applied Engineering, 9 a.m. to 11 p.m. 7 days. Or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 5% sales tax. Add \$10.00 outside U.S.A.

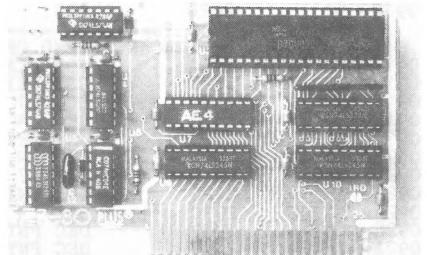
AE APPLIED ENGINEERING™

The Apple enhancement experts

(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

With the Z-80 Plus, run over 5000 new CP/M® programs.



*Now, get two computers
in one, and all the
advantages of both.*

Enter the CP/M world with the new Z-80 Plus card from Applied Engineering, and introduce your Apple IIe, IIcs or II+ to thousands of CP/M programs. Only the Z-80 Plus comes standard with the new 5.1 software, the most advanced system ever for running CP/M programs.

The new 5.1 boasts advanced features like built-in disk emulation for popular memory expansion boards, boosting both system speed and storage capacity. And menu-driven utilities that let you get to work faster. The Z-80 Plus also lets you run older CP/M programs—all the way down to Version 1.6 (2.2 is the most popular).

The new Z-80 Plus is the only card on the market capable of accessing more than 64K. If you have an extended 80 column card, all 128K is usable. And if you have RamWorks, RamFactor, o-Ram, o-Ram Plus, or an Apple memory card in your IIe, IIcs or II+, up to 1088K is available.

Each Z-80 Plus comes with our CP/M Ram Drive software, enabling most memory cards to work as a high-speed Ram disk so CP/M software can run up to *twenty times faster*. So packages like WordStar, dBASE II, Turbo Pascal and most others run at blinding speed.

Simply plug the Z-80 Plus into any slot in your Apple. You'll get the benefits of two computers in one—all at an unbelievably low price (only \$149!).

- Fully compatible with ALL CP/M software
- Fully compatible with 5¼" and 3½" disk drives as well as most hard disks including the Sider
- Fully compatible with Microsoft disks (no pre-boot required)
- Specifically designed for high speed operation
- Runs WordStar, dBASE II, Turbo Pascal, and ALL other CP/M software with *no pre-boot*
- Semi-custom I.C. and low parts count allows Z-80 Plus to fly through CP/M programs with extremely low power consumption (we use the Z80B3)
- Does EVERYTHING other Z-80 boards do, *plus* Z-80 interrupts
- Five-year warranty

To get your Z-80 Plus or for more information see your dealer, or call 214-241-6060 9 a.m. to 11 p.m. seven days, or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 5% sales tax. Add \$10.00 outside U.S.A.

AE Applied Engineering™
P.O. Box 798, Carrollton, TX 75006
(214) 241-6060

CP/M is a registered trademark of Digital Research.

```

0903- 20 0A 09 2720 SCAN.TO.END.OF.PAGE
0906- 20 13 09 2730 JSR SCAN.32.LINES
0909- 60 2740 JSR SCAN.TO.END.OF.CURRENT.ENTRY
2750 RTS
2760 *-----
2770 SCAN.32.LINES
090A- A2 20 2780 LDY #32
090C- 20 2D 09 2790 .1 JSR SCAN.TO.NEXT.LINE
090F- CA 2800 DEX
0910- D0 FA 2810 BNE .1
0912- 60 2820 RTS
2830 *-----
2840 SCAN.TO.END.OF.CURRENT.ENTRY
0913- 20 2D 09 2850 .1 JSR SCAN.TO.NEXT.LINE
0916- 20 33 09 2860 JSR NEXT.CHAR
0919- 90 F8 2870 BCC .1 ...not <RETURN> or end-of-text
091B- F0 0F 2880 BEQ .4 ...found end-of-text
091D- 20 33 09 2890 .2 JSR NEXT.CHAR
0920- F0 0A 2900 BEQ .4 ...found end-of-text
0922- B0 F9 2910 BCS .2 ...found another <RETURN>
0924- A5 19 2920 LDA PNTR Backup the text pointer
0926- D0 02 2930 BNE .3 to the <RETURN>
0928- C6 1A 2940 DEC PNTR+1
092A- C6 19 2950 .3 DEC PNTR
092C- 60 2960 .4 RTS
2970 *-----
2980 SCAN.TO.NEXT.LINE
092D- 20 33 09 2990 .1 JSR NEXT.CHAR
0930- 90 FB 3000 BCC .1 ...not end-of-text or <RETURN>
0932- 60 3010 RTS
3020 *-----
3030 * Get next character from text buffer
3040 * Return .CC. if not <RETURN> or end-of-text
3050 * .CS. and .EQ. if end-of-text
3060 * .CS. and .NE. if <RETURN>
3070 *-----
3080 NEXT.CHAR
0933- 38 3090 SEC Return .CS. if <eot> or <return>
0934- A0 00 3100 LDY #0
0936- B1 19 3110 LDA (PNTR),Y
0938- F0 0B 3120 BEQ .2 ...it is end-of-text
093A- E6 19 3130 INC PNTR
093C- D0 02 3140 BNE .1
093E- E6 1A 3150 INC PNTR+1
0940- C9 8D 3160 .1 CMP #$8D
0942- F0 02 3170 BEQ .3
0944- 18 3180 CLC
0945- 60 3190 .2 RTS
0946- C8 3200 .3 INY Return .CS. and .NE. if <RETURN>
0947- 60 3210 RTS
3220 *-----
0948- 3230 SPAGES .BS 24
3240 *-----

```

Apple Assembly Line (ISSN 0889-4302) is published monthly by S-C SOFTWARE CORPORATION, P.O. Box 280300, Dallas, Texas 75228. Phone (214) 324-2050. Subscription rate is \$18 per year in the USA, sent Bulk Mail; add \$3 for First Class postage in USA, Canada, and Mexico; add \$14 postage for other countries. Back issues are \$1.80 each (other countries inquire for postage). A subscription to the newsletter and the Monthly Disk containing all source code is \$64 per year in the US, Canada and Mexico, and \$87 to other countries.

All material herein is copyrighted by S-C SOFTWARE CORPORATION, all rights reserved. (Apple is a registered trademark of Apple Computer, Inc.)